Operating Instruction Manual

DTM for Hilscher DeviceNet Slave Devices

Configuration of Hilscher Slave Devices

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC091003OI08EN | Revision 8 | English | 2013-09 | Released | Public

Table of Contents

1 INTRODUCTION 5
1.1 About this Manual 5
1.1.1 Descriptions of the Dialog Panes 5
1.1.2 Online Help 6
1.1.3 List of Revisions 6
1.1.4 Conventions in this Manual 7
1.2 Legal Notes 8
1.2.1 Copyright 8
1.2.2 Important Notes 8
1.2.3 Exclusion of Liability 9
1.2.4 Warranty 9
1.2.5 Export Regulations 10
1.2.6 Registered Trademarks 10
1.3 About DeviceNet Slave DTM 11
1.3.1 Requirements 12
1.4 Dialog Structure of the DeviceNet Slave DTM 13
1.4.1 General Device Information 14
1.4.2 Navigation Area 14
1.4.3 Dialog Panes 15
1.4.4 OK, Cancel, Apply and Help 16
1.4.5 Table Lines 16
1.4.6 Status Bar 17

2 GETTING STARTED 18
2.1 Configuration Steps 18
2.1.1 Slave DTM at the Root-Line (Stand-Alone Slave) 18
2.1.2 Slave DTM at the Master busline 20

3 SETTINGS 24
3.1 Overview Settings 24
3.2 Settings for Driver and Device Assignment 25
3.3 Driver 27
3.3.1 Verify or adapt Driver Settings 27
3.3.2 cifX Device Driver 29
3.3.3 netX Driver 29
3.3.4 Configuring netX Driver 30
3.3.5 netX Driver - USB/RS232 Connection 31
3.3.6 netX Driver - TCP/IP Connection 34
3.4 Device Assignment 37
3.4.1 Scanning for Devices 37
3.4.2 Selecting the Device (with or without firmware) 40
3.4.3 Selecting the Device once more (with Firmware) 41

© Hilscher, 2009-2013

3.5 Firmware Download 43

4 CONFIGURATION 48

4.1 Overview Configuration 48

4.2 Configuring Slave Parameters 49

4.2.1 Slave DTM at the Root-Line (Stand-Alone Slave) 49

4.2.2 Slave DTM at the Master busline 50

4.3 General Settings 51

4.3.1 Enable Address Switch 52

4.4 Connection Configuration 53

4.4.1 Connection Types 53

4.5 Parameter 61

4.6 Device Settings 62

4.6.1 Start of Bus Communication 63

4.6.2 Application Monitoring 63

5 DEVICE DESCRIPTION 64

5.1 About Device Description 64

5.2 Device 64

5.3 EDS 65

6 ONLINE FUNCTIONS 66

6.1 Connecting/Disconnecting Device 66

6.2 Download Configuration 67

7 DIAGNOSIS 68

7.1 Overview Diagnosis 68

7.2 General Diagnosis 69

7.3 Firmware Diagnosis 71

8 EXTENDED DIAGNOSIS 72

8.1 Overview Extended Diagnosis 72

8.2 Task Information 73

8.3 IniBatch Status 74

8.4 DNS_FAL 75

8.4.1 DNS Common Status 75

8.4.2 AP Commands Counter 76

8.4.3 CAN Command Counter 76

8.4.4 Timer Counter 77

8.5 CAN_DL 78

8.5.1 AP Commands Counter 78

8.5.2 CAN Driver Status 79

9 TOOLS 80

© Hilscher, 2009-2013

9.1 Overview Tools 80
9.2 Packet Monitor 81
9.2.1 Sending Packet 82
9.2.2 Receiving Packet 83
9.2.3 Example – „Get DPM I/O Information“ 84
9.3 IO Monitor 89

10 ERROR CODES 90
10.1 Error Code Definition 90
10.2 Overview Error Codes 91
10.3 General Hardware Error Codes 92
10.3.1 RCX General Task Errors 92
10.3.2 RCX Common Status & Errors Codes 93
10.3.3 RCX Status & Error Codes 94
10.4 ODM Error Codes 95
10.4.1 General ODM Error Codes 95
10.4.2 General ODM Driver Error Codes 96
10.4.3 cifX Driver Specific ODM Error Codes 97
10.5 Error Codes cifX Device Driver and netX Driver 100
10.5.1 Generic Error Codes 100
10.5.2 Generic Driver Error Codes 101
10.5.3 Generic Device Error Codes 102
10.6 Error Codes netX Driver 103
10.6.1 CIFX API Transport Error Codes 103
10.6.2 CIFX API Transport Header State Error Codes 103
10.7 ODM Error Codes DBM V4 104

11 APPENDIX 108
11.1 User Rights 108
11.1.1 Settings 108
11.1.2 Configuration 108
11.2 References 109
11.3 List of Figures 109
11.4 List of Tables 110
11.5 Glossary 111
11.6 Contacts 113

© Hilscher, 2009-2013

1 Introduction

1.1 About this Manual

This manual provides information on how to set and configure the device parameters of a netX based DeviceNet Slave device within a FDT Framework using the DeviceNet Slave DTM, and what can be read from the diagnosis panes.

In a network project the DeviceNet Slave DTM can be dropped:

- to the root line (Stand-Alone Slave),
- or to the Master busline of a DeviceNet Master DTM.

Note: This manual edition first provides information for the case if the DeviceNet Slave DTM is dropped to the root line (Stand-Alone Slave) and in addition important information for the case if the DeviceNet Slave DTM is dropped to the Master busline of a DeviceNet Master DTM.

1.1.1 Descriptions of the Dialog Panes

The table below gives an overview for the individual dialog panes descriptions:

Section	Subsection	Manual Page
Settings	*Overview Settings*	24
	Driver	27
	Device Assignment	37
	Firmware Download	43
Configuration	*Overview Configuration*	48
	General Settings	51
	Connection Configuration	53
	Poll Connection	54
	Change of State Connection	56
	Cyclic Connection	58
	Bit-Strobe Connection	59
	Parameter	61
	Device Settings	62
Device Description	*Device*	64
	EDS	65
Diagnosis	*Overview Diagnosis*	68
	General Diagnosis	69
	Firmware Diagnosis	71
Extended Diagnosis	*Overview Extended Diagnosis*	72
Tools	*Overview Tools*	80
	Packet Monitor	81
	IO Monitor	89

Table 1: Descriptions Dialog Panes

© Hilscher, 2009-2013

1.1.2 Online Help

The generic DeviceNet Slave DTM contains an integrated online help facility.

- To open the online help, click on **Help** or press **F1**.
-

1.1.3 List of Revisions

Index	Date	Version	Component	Chapter	Revision
7	13-04-16	1.8.x.x, 1.8.x.x	DevNetSlaveDTM.dll DevNetSlaveGUI.ocx	All, 1.3.1	Revised. Section *Requirements*, Windows 8 added.
8	13-09-16	1.8.x.x, 1.8.x.x	DevNetSlaveDTM.dll DevNetSlaveGUI.ocx	All; 2.1; 3.2, 3.3, 3.5	Note about the configuration of 2-channel devices added. Sections *Configuration Steps*, *Settings for Driver and Device Assignment* and *Driver* updated. Section *Firmware Download* updated.

© Hilscher, 2009-2013

1.1.4 Conventions in this Manual

Notes, operation instructions and results of operation steps are marked as follows:

Notes

Important: <important note>

Note: <note>

<note, where to find further information>

Operation Instructions

1. <instruction>
2. <instruction>

or

➢ <instruction>

Results

⇨ <result>

© Hilscher, 2009-2013

1.2 Legal Notes

1.2.1 Copyright

© Hilscher, 2009-2013, Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

1.2.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

1.2.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

© Hilscher, 2009-2013

1.2.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different counters, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

1.2.6 Registered Trademarks

Windows® XP, Windows® Vista, Windows® 7 and Windows® 8 are registered trademarks of Microsoft Corporation.

DeviceNet™ is a trademark of ODVA (Open DeviceNet Vendor Association, Inc).

CODESYS® is a registered trademark of 3S-Smart Software Solutions GmbH, Deutschland.

All other mentioned trademarks are property of their respective legal owners.

© Hilscher, 2009-2013

1.3 About DeviceNet Slave DTM

You can use the DeviceNet Slave DTM to configure DeviceNet Slave devices within a FDT Framework and to view the device diagnosis.

In a network project the DeviceNet Slave DTM can be dropped:

- to the root line (Stand-Alone Slave),
- or to the Master busline of a DeviceNet Master DTM.

Slave DTM at the Root-Line (Stand-Alone Slave)

Insert the DeviceNet Slave DTM to the root busline if you intend to configure only one single DeviceNet Slave device.

Slave DTM at the Master Busline

Insert the DeviceNet Slave DTM to the Master busline of the DeviceNet Master DTM if within the network project the Slave device and the Master device are used, i. e. if you intend to configure the Slave device and the Master device.

2-Channel Devices

Important: For a 2-channel device consecutively channel 1 or channel 2 must be assigned to the DTM and each must be configured individually.

© Hilscher, 2009-2013

1.3.1 Requirements

System Requirements

- PC with 1 GHz processor or higher
- Windows® XP SP3, Windows® Vista (32 bit) SP2, Windows® 7 (32 bit) SP1, Windows® 7 (64 bit) SP1, Windows® 8 (32 bit) or Windows® 8 (64 bit)
- Administrator privilege required for installation
- Internet Explorer 5.5 or higher
- RAM: min. 512 MByte, recommended 1024 MByte
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse

Note: If the project file is saved and opened again or if it is used on another PC, the system requirements must match. Particularly the DTM must be installed on the used PC.

Restriction

Touch screen is not supported.

Requirements DeviceNet Slave DTM

To configure the DeviceNet Slave device with the DeviceNet Slave DTM the following requirements have to be accomplished:

- Completed hardware installation of a netX based DTM-compatible DeviceNet Slave device, inclusive loaded firmware and loaded configuration file
- Installed FDT/DTM V 1.2 compliant frame application
- Installed DeviceNet Master DTM (If Slave DTM is dropped to the Master busline.)
- Loaded DTM in the Device Catalog of the FTD Framework

Note: If the DeviceNet Slave DTM and the DeviceNet Slave device are installed on the same PC, the **cifX Device Driver** must be installed on that PC, as you can connect the DTM to the device.

For more information to the hardware installation, please refer to the corresponding **User Manual** of your device.

© Hilscher, 2009-2013

1.4 Dialog Structure of the DeviceNet Slave DTM

The graphical user interface of the DTM is composed of different areas and elements listed hereafter:

1. A header area containing the **General Device Information**,
2. The **Navigation Area** (area on the left side),
3. The **Dialog Pane** (main area on the right side),
4. **OK**, **Cancel**, **Apply**, **Help**,
5. The **Status Line** containing information e. g. the online-state of the DTM.

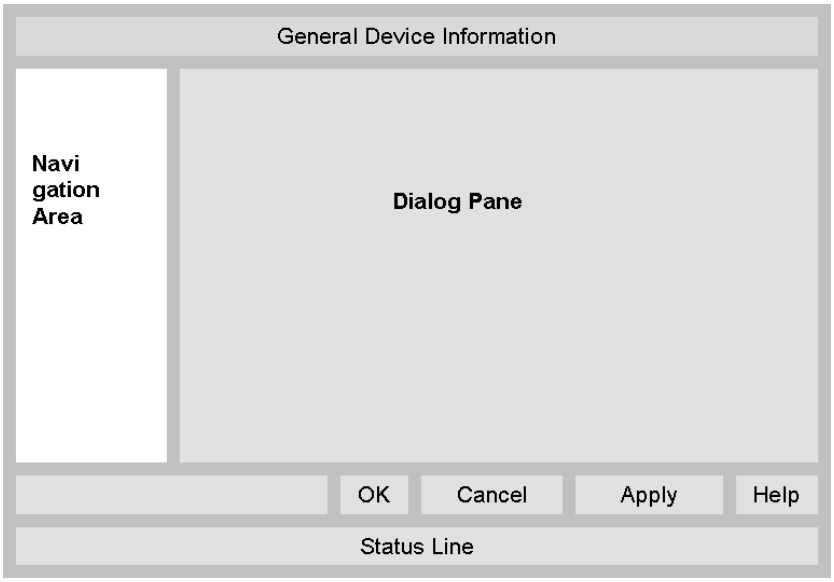


Figure 1: Dialog Structure of the DeviceNet Slave DTM

© Hilscher, 2009-2013

1.4.1 General Device Information

Parameter	Meaning
IO Device	Name of the device
Vendor	Vendor name of the device
Device ID	Identification number of the device
Vendor ID	Identification number of the vendor

Table 2: General Device Information

1.4.2 Navigation Area

The **Navigation Area** contains folders and subfolders to open the dialog panes of the DTM.

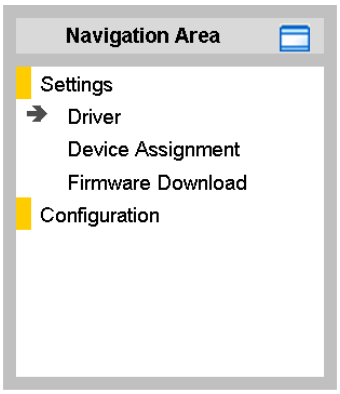


Figure 2: Navigation Area

➢ Select the required folder and subfolder.

➢ The corresponding Dialog pane is displayed.

Hide / display Navigation

	Hiding the navigation area (above right side).
Show navigation area	Opening the navigation area (below left side).

© Hilscher, 2009-2013

1.4.3 Dialog Panes

At the dialog pane the **Settings**, **Configuration**, **Diagnosis**/**Extended Diagnosis** or the **Tools** panes are opened via the corresponding folder in the navigation area.

Settings	
Driver	To establish a connection from the DeviceNet Slave DTM to the DeviceNet Slave device, on the pane **Driver** you can verify if the default driver is checked and respectively check another driver or multiple drivers. For further information, refer to section *Driver* on page *27*.
Device Assignment	On the pane **Device Assignment** you select the device and assign the device to the driver. For further information, refer to section *Device Assignment* on page *37*.
Firmware Download	The dialog on the pane **Firmware Download** is used to load a new firmware into the device. A detailed description can be found in section *Firmware Download* on page *43*.
Configuration	
General	On the page **General Settings** the MAC-ID can be read and the parameters 'UCMM ', 'Fragmentation Timeout' or 'Verify Device ID' can be preset. Further information to this you find in section *General Settings* on page *51*.
…Connection	On the page **Connection Configuration** a connection type can be selected and configured. Further information to this you find in section *Connection Configuration* on page *53*.
Parameters	In the **Parameter** dialog the parameter data of the device can be edited. A detailed description you find in section *Parameter* on page *61.*
Device Settings	At the **Device Settings** pane device related settings can be made. For further information, refer to section *Device Settings* on page 62.
Device Description	
Device	The **Device Info** pane contains the manufacturer information about the device. Further information to this you find in section *Device* on page *64.*
EDS	By use of the **EDS-Viewer** an EDS file can be searched through. Further information to this you find in section *EDS* on page 65*.*
Diagnosis	
Diagnosis/ Extended Diagnosis	At the **Diagnosis** panes information can be read for troubleshooting. For further information, refer to section *Overview Diagnosis* on page 68.
Tools	
Packet Monitor/ IO Monitor	Under **Tools** the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes. For further information, refer to section *Packet Monitor* on page *81* or section *IO Monitor* on page 89.

Table 3: Overview Dialog Panes

Note: Accessing the **Diagnosis** panes of the DeviceNet Slave DTM requires an online connection from the DeviceNet Slave DTM to the DeviceNet Slave DTM.

For further information, refer to section *Connecting/Disconnecting Device* on page 66.

© Hilscher, 2009-2013

1.4.4 OK, Cancel, Apply and Help

OK, **Cancel**, **Apply** and **Help** you can use as described hereafter.

	Meaning
OK	To confirm your latest settings, click **OK**. All changed values will be applied on the frame application database. *The dialog then closes.*
Cancel	To cancel your latest changes, click **Cancel**. Answer to the safety query **Configuration data has been changed. Do you want to save the data?** by **Yes**, **No** or **Cancel**. **Yes**: The changes are saved or the changed values are applied on the frame application database. *The dialog then closes.* **No**: The changes are not saved or the changed values are not applied on the frame application database. *The dialog then closes.* **Cancel**: *Back to the DTM.*
Apply	To confirm your latest settings, click **Apply**. All changed values will be applied on the frame application database. *The dialog remains opened.*
Help	To open the DTM online help, click **Help**.

Table 4: OK, Cancel, Apply and Help

1.4.5 Table Lines

In the DTM dialog pane table lines can be selected, inserted or deleted.

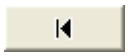
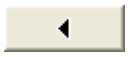
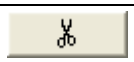
	Meaning
	To select the first line of a table use **First Line**.
	To select the previous line of a table use **Previous Line**.
	To select the next line of a table use **Next Line**.
	To select the last line of a table use **Last Line**.
	Create a new Line inserts new lines into the table.
	**Delete selected Line** deletes the selected line from the table.

Table 5: Selecting, inserting, deleting Table Line

© Hilscher, 2009-2013

1.4.6 Status Bar

The **Status Bar** displays information about the current state of the DTM. The current activity, e.g. download, is signaled graphically via icons in the status bar.

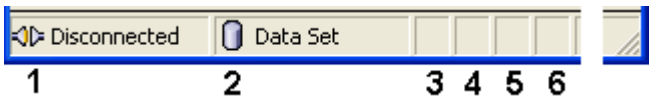
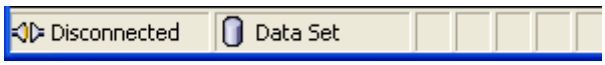


Figure 3: Status Bar – Status Fields 1 to 6

Status Field	Icon / Meaning	
1	**DTM Connection States**	
		Connected: Icon closed = Device is online
		Disconnected: Icon opened = Device is offline
2	**Data Source States**	
		Data set: The displayed data are read out from the instance data set (database).
		Device: The displayed data are read out from the device.
3	**States of the instance Date Set**	
		Valid Modified: Parameter is changed (not equal to data source).
4	**Changes directly made on the Device**	
		Load/configure diagnosis parameters: Diagnosis is activated.
6	**Device Diagnosis Status**	
		Save operation succeeded: The save operation has been successful.
		Further messages due to successful handling of device data.
		Firmware Download: Firmware Download is running
		Save operation failed: The save operation has failed.
		Further fail operation messages due to incorrect communication due to malfunction in the field device or its peripherals.

Table 6: Status Bar Icons [1]

Offline State



Save operation succeeded

Firmware Download

Firmware Download successful

Online State and Diagnosis

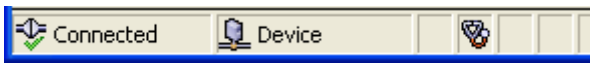


Figure 4: Status Bar Display Examples

© Hilscher, 2009-2013

2 Getting started

2.1 Configuration Steps

The following overview provides to you the step sequence on how to configure a netX based DeviceNet Master device with DeviceNet Master DTM as it is typical for many cases. At this time it is presupposed that the hardware installation was done.

The overview lists all the steps in a compressed form. For detailed descriptions of each step refer to the sections noted in the column *For detailed information see section*.

The following two cases are considered:

- Slave DTM at the Root-Line (Stand-Alone Slave)
- and Slave DTM at the Master busline.

2-Channel Devices

Important: For a 2-channel device consecutively channel 1 or channel 2 must be assigned to the DTM and each must be configured individually.

2.1.1 Slave DTM at the Root-Line (Stand-Alone Slave)

#	Step	Short Description	For detailed information see section	Page
1	Load device catalog	Depending of the FDT Container: For netDevice: - select **Network > Device Catalog**, - select **Reload Catalog**.	*(See Operating Instruction Manual netDevice and netProject)*	-
2	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select **File > New** or **File > Open**.	*(See Operating Instruction Manual of the Frame Application)*	-
3	Insert Slave into configuration	Depending of the FDT Container: For netDevice: - in the Device Catalog under **Gateway / Stand-Alone Slave** click to the Slave, - and insert the Slave via drag and drop **to the root line** in the network view.	*(See Operating Instruction Manual netDevice and netProject)*	-
4	Open the Slave DTM configu-ration dialog	Open the Slave DTM configuration dialog. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed.	-	-

© Hilscher, 2009-2013

#	Step	Short Description	For detailed information see section	Page
5	Verify or adapt Driver Settings	In the Slave DTM configuration dialog: - select **Settings > Driver**. • **Note!** For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.Use the **cifX Device Driver** if the DeviceNet Slave DTM is installed on the same PC as the DeviceNet Slave device. • Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the DeviceNet Slave DTM to the DeviceNet Slave device. • The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS. To search for devices you can check one or multiple drivers simultaneously. - Verify that the default driver is checked. - If necessary, check another driver or multiple drivers.	*Settings for Driver and Device Assignment* and *Driver*	25 27
6	Configure Driver	If you use the **netX Driver**, you respectively must configure it. For **netX Driver and communication via TCP/IP** set the IP address of the device: - Select **Settings > Driver > netX Driver > TCP Connection**. - Via add an IP range. - Under **IP Address** enter the IP Address of the device or an IP range. - Click **Save**. Adjust the driver parameters **netX Driver USB/RS232** only if they differ from the default settings. **Note!** • The **cifX Device Driver** requires no configuration. • The configuration of the **3SGateway Driver for netX (V3.x)** is carried out via the CODESYS surface.	*Configuring netX Driver*	30
7	Assign Slave device (with or without firmware)	Assign the device to this driver. In the Slave DTM configuration dialog: - select **Settings > Device Assignment**, - select a Slave device (with or without firmware), - therefore check the appropriate checkbox, - select **Apply**.	*Selecting the Device (with or without firmware)*	40
8	Select and download firmware	If not yet a firmware was loaded to the device. In the Slave DTM configuration dialog: - select **Settings > Firmware Download**, - select **Browse..**, - select a firmware file, - select **Open**, - select **Download** and **Yes**.	*Firmware Download*	43

© Hilscher, 2009-2013

#	Step	Short Description	For detailed information see section	Page
9	Assign Slave device once more (with firmware and system chanal) *For repeated download this step is omitted.*	In the Slave DTM configuration dialog: - select **Settings > Device Assignment**, - select **Scan**, - select the Slave device (with loaded firmware and defined system channel), - therefore check the appropriate checkbox, - select **Apply**, - close the Slave DTM configuration dialog via **OK**.	*Selecting the Device once more (with* Firmware)	41
10	Configure Slave device	Configure the Slave device. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed. In the Slave DTM configuration dialog: - select **Configuration >General**, - set UCMM and Fragmentation Timeout, - select **Configuration > Connection**, - configure the device connection, - select **Configuration >Parameter**, - set the parameter data of the device, - select **Configuration > Device Settings**, - set the Device Settings. - close the Slave DTM configuration dialog via **OK**.	*Configuring Slave Parameter* *General Settings* *Connection Configuration* *Parameter* *Device Settings*	49 51 53 61 62
11	Save project	Depending of the frame application. For the configuration software: - select **File > Save**.	*(See Operating Instruction Manual of the Frame Application)*	-
12	Connect Slave device	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Connect**.	*Connecting/Disconnecting Device*	66
13	Download Configuration	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Download**.	*Download Configuration*	67
14	Diagnosis	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Diagnosis**. - The Slave DTM diagnosis dialog is displayed. (1) Check whether the communication is OK: **Diagnosis** > **General Diagnosis** > **Device status "Communication"** must be green! (2) **„Communication“** is green: Open thel IO Monitor and test the input or output data. (3) **„Communication“** is not green: Use Diagnosis and Extended diagnosis for troubleshooting. - close the Slave DTM diagnosis dialog via **OK**.	*Overview Diagnosis*	68
15	IO Monitor	Depending of the FDT Container: For netDevice: - right click to the device icon of the Slave, - select **Diagnosis**, - select **Tools > IO Monitor**. - Check the input or output data.	*IO Monitor*	89
16	Disconnect	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Disconnect**.	*Connecting/Disconnecting Device*	66

Table 7: Getting started - Configuration Steps (Slave DTM at the Root-Line (Stand-Alone Slave))

2.1.2 Slave DTM at the Master busline

© Hilscher, 2009-2013

#	Step	Short Description	For detailed information see section	Page
1	Load device catalog	Depending of the FDT Container: For netDevice: - select **Network > Device Catalog**, - select **Reload Catalog**.	*(See Operating Instruction Manual netDevice and netProject)*	-
2	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select **File > New** or **File > Open**.	*(See Operating Instruction Manual of the Frame Application)*	-
3	Insert Master or Slave into configuration	For netDevice: - in the Device Catalog click to the Master, - and insert the device via drag and drop **to the root line** in the network view, - in the Device Catalog click to the Slave, - and insert the device via drag and drop **to the Master bus line** in the network view.	*(See Operating Instruction Manual netDevice and netProject)*	-
4	Open the Slave DTM configu-ration dialog	Open the Slave DTM configuration dialog. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed.	-	-
5	Verify or adapt Driver Settings	In the Slave DTM configuration dialog: - select **Settings > Driver**. **Note!** For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver. • Use the **cifX Device Driver** if the DeviceNet Slave DTM is installed on the same PC as the DeviceNet Slave device. • Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the DeviceNet Slave DTM to the DeviceNet Slave device. • The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS. To search for devices you can check one or multiple drivers simultaneously. - Verify that the default driver is checked. - If necessary, check another driver or multiple drivers.	*Settings for Driver and Device Assignment* and *Driver*	25 27
6	Configure Driver	If you use the **netX Driver**, you respectively must configure it. For **netX Driver and communication via TCP/IP** set the IP address of the device: - Select **Settings > Driver > netX Driver > TCP Connection**. - Via add an IP range. - Under **IP Address** enter the IP Address of the device or an IP range. - Click **Save**. Adjust the driver parameters **netX Driver USB/RS232** only if they differ from the default settings. **Note!** • The **cifX Device Driver** requires no configuration. • The configuration of the **3SGateway Driver for netX (V3.x)** is carried out via the CODESYS surface.	*Configuring netX Driver*	30

© Hilscher, 2009-2013

#	Step	Short Description	For detailed information see section	Page
7	Assign Slave device (with or without firmware)	Assign the device to this driver. In the Slave DTM configuration dialog: - select **Settings > Device Assignment**, - select a Slave device (with or without firmware), - therefore check the appropriate checkbox, - select **Apply**.	*Selecting the Device (with or without firmware)*	40
8	Select and download firmware	If not yet a firmware was loaded to the device. In the Slave DTM configuration dialog: - select **Settings > Firmware Download**, - select **Browse..**, - select a firmware file, - select **Open**, - select **Download** and **Yes**.	*Firmware Download*	43
9	Assign Slave device once more (with firmware and system chanal) *For repeated download this step is omitted.*	In the Slave DTM configuration dialog: - select **Settings > Device Assignment**, - select **Scan**, - select the Slave device (with loaded firmware and defined system channel), - therefore check the appropriate checkbox, - select **Apply**, - close the Slave DTM configuration dialog via **OK**.	*Selecting the Device once more (with* Firmware)	41
10	Configure Slave device	Configure the Slave device. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed. In the Slave DTM configuration dialog: - select **Configuration >General**, - set UCMM and Fragmentation Timeout, - select **Configuration > Connection**, - configure the device connection, - select **Configuration >Parameter**, - set the parameter data of the device, - select **Configuration > Device Settings**, - set the Device Settings. - close the Slave DTM configuration dialog via **OK**.	*Configuring Slave Parameter* *General Settings* *Connection Configuration* *Parameter* *Device Settings*	49 51 53 61 62
11	Configure Master device	Configure the Master device via the DeviceNet Master DTM netX.	*(See Operating Instruction Manual DTM for DeviceNet Master devices)*	-
12	Save project	Depending of the frame application. For the configuration software: - select **File > Save**.	*(See Operating Instruction Manual of the Frame Application)*	-
13	Connect Slave device	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Connect**.	*Connecting/Disconnecting Device*	66
14	Download Configuration	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Download**.	*Download Configuration*	67
15	Diagnosis	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Diagnosis**. - The Slave DTM diagnosis dialog is displayed. (1) Check whether the communication is OK: **Diagnosis** > **General Diagnosis** > **Device status "Communication"** must be green! (2) **„Communication“** is green: Open thel IO Monitor and test the input or output data. (3) **„Communication“** is not green: Use Diagnosis and Extended diagnosis for troubleshooting. - close the Slave DTM diagnosis dialog via **OK**.	*Overview Diagnosis*	68

© Hilscher, 2009-2013

#	Step	Short Description	For detailed information see section	Page
16	IO Monitor	Depending of the FDT Container: For netDevice: - right click to the device icon of the Slave, - select **Diagnosis**, - select **Tools > IO Monitor**. - Check the input or output data, - close the IO Monitor dialog via **OK**.	*IO Monitor*	89
17	Disconnect	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select **Disconnect**.	*Connecting/Disconnecting Device*	66

Table 8: Getting started - Configuration Steps (Slave DTM at the Master busline)

© Hilscher, 2009-2013

3 Settings

3.1 Overview Settings

Settings Dialog Panes

The table below gives an overview for the individual **Settings** dialog panes descriptions:

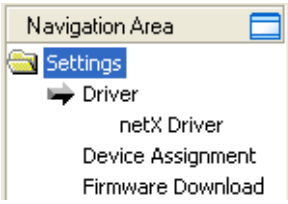
DeviceNet Slave DTM	Folder Name / Section	Subsection	Manual Page
 *Navigation Area – Settings (Example)* *Additional drivers can be displayed.*	*Driver*		27
		Verify or adapt Driver Settings	27
		cifX Device Driver	29
		netX Driver	29
		Configuring netX Driver	30
	Device Assignment		37
		Scanning for Devices	37
		Scanning for all Devices or for suitable only	39
		Selecting the Device (with or without firmware)	40
		Selecting the Device once more (with Firmware)	41
	Firmware Download		43

Table 9: Descriptions of the Dialog Panes Settings

Note: To edit the **Settings** dialog panes you need *User Rights* for “Maintenance”.

Notice the descriptions in the section *Settings for Driver and Device Assignment on* page 25.

To access to the online help with the descriptions of the drivers:

- Select **Settings > Driver >** [Name *of the assigned driver].*
- Press the **F1** key..

© Hilscher, 2009-2013

3.2 Settings for Driver and Device Assignment

Important: For a 2-channel device consecutively channel 1 or channel 2 must be assigned to the DTM.

The following steps are needed to establish a connection from the DeviceNet Slave DTM to the DeviceNet Slave device:

Verify or adapt Driver Settings

Verify the Driver Settings and adapt them if necessary.

1. Open the DTM configuration dialog.

➢ In the FDT container **netDevice** double click to the DeviceNet Slave device icon.

2. Verify that the default driver is checked and respectively check another or multiple drivers.

➢ Select **Settings > Driver**.

Note! For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.

- Use the **cifX Device Driver** if the DeviceNet Slave DTM is installed on the same PC as the DeviceNet Slave device.
- Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the DeviceNet Slave DTM to the DeviceNet Slave device.
- The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS.

To search for devices on the network you can check one or multiple drivers simultaneously.

➢ Verify that the default driver for your device is checked.

➢ If necessary, check another driver or multiple drivers.

Configure Driver

Note!

- The **cifX Device Driver** requires no configuration.
- The configuration of the **3SGateway Driver for netX (V3.x)** is carried out via the CODESYS surface.

If you use the **netX Driver**, you respectively must configure it.

3. Configure the **netX Driver** if necessary.

For the driver **netXDriver** an individual driver dialog window can be opened where you can configure the driver.

➢ Select **Settings > Driver** > **netX Driver**.

➢ For netX Driver and communication via TCP/IP set the IP address of the device.

Adjust the driver parameters **netX Driver USB/RS232** only if they differ from the default settings.

© Hilscher, 2009-2013

Assigning the Slave device to the DTM

4. Scan for and select the devices (with or without firmware).

➢ Select **Settings > Device Assignment**.

➢ Under **Device selection** select *suitable only* or *all* and then **Scan**.

➢ In the table check the required devices.

➢ Select **Apply**.

Select and download the Firmware

5. If not yet a firmware was loaded to the device, select and download the firmware.

➢ Select **Settings > Firmware Download**.

➢ Select and download the firmware via **Download**.

➢ Select **Apply**.

6. Scan for and select the devices (with firmware and defined system channel) once more.

   For repeated download this step is omitted.

➢ Select **Settings > Device Assignment**.

➢ Select **Scan**.

➢ In the table check the required devices.

7. Close the DTM configuration dialog via **OK**.

Connecting the Device

8. In **netDevice** put a right-click on the DeviceNet Slave device icon.
9. Select the **Connect** command from the context menu.

↬ In the network view the device description at the device icon of the Slave is displayed with a green colored background. The DeviceNet Slave device now is connected to the DeviceNet Slave DTM via an online connection.

Further Information

For descriptions about these steps refer to the sections following hereafter.

© Hilscher, 2009-2013

3.3 Driver

The **Driver** dialog pane displays the drivers to be used for a DeviceNet Slave DTM to establish a device communication connection.

Note! A **default driver** is set in the configuration software.

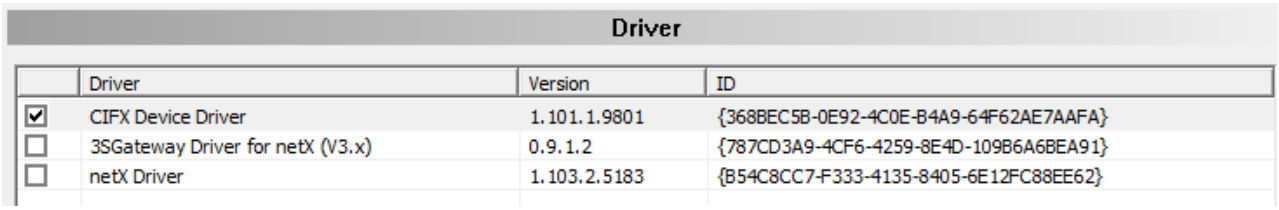


Figure 5: Default Driver ‚cifX Device Driver' for PC cards cifX

Parameter	Meaning
Driver	Name of the driver (for more details see descriptions hereafter)
Version	ODMV3 Version of the respective driver
ID	ID of the driver (driver identification)

Table 10: Driver Selection List Parameters

To establish a connection from the DeviceNet Slave DTM to the DeviceNet Slave device, verify if the default driver is checked and respectively check another driver or multiple drivers.

3.3.1 Verify or adapt Driver Settings

Proceed as follows:

1. Select **Settings > Driver** in the navigation area.

↬ The **Driver** dialog pane is displayed with the available drivers and the setting for the default driver.

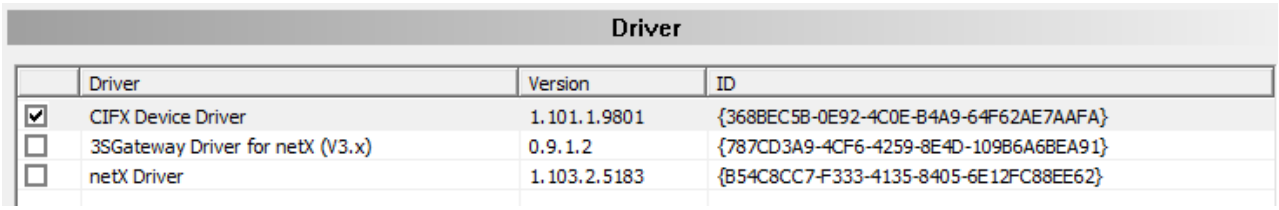


Figure 6: Default Driver ‚cifX Device Driver' for PC cards cifX (example)

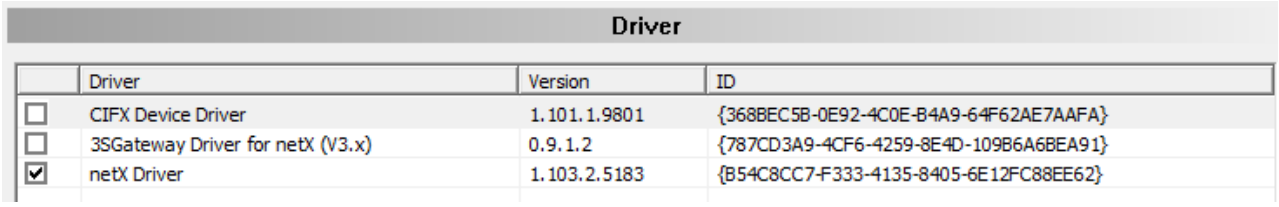


Figure 7: Default Driver ‚netX Driver' for Hilscher devices except for PC cards cifX (example)

2. Verify that the default driver is checked.

➢ Verify that the default driver for your device is checked.

© Hilscher, 2009-2013

Default Driver (Pre-settings in the Configuration Software): For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.

3. Respectively check another driver.

Note! The driver used for the connection from the DeviceNet Slave DTM to the DeviceNet Slave device must be supported by the device and must be available for the device.

- Use the **cifX Device Driver** if the DeviceNet Slave DTM is installed on the same PC as the DeviceNet Slave device.
- Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the DeviceNet Slave DTM to the DeviceNet Slave device.
- The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS. The version V3.x refers to the driver version defined by 3S-Smart Software Solutions GmbH.

➢ Check the checkbox for the driver in the selection list.

4. Respectively check multiple drivers.

To search for devices on the network you can check multiple drivers simultaneously.

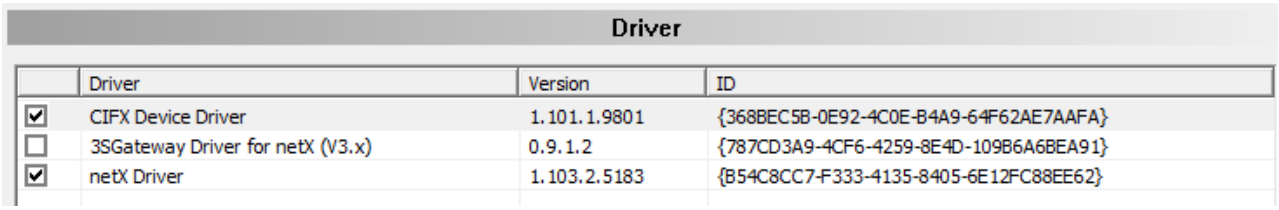


Figure 8: Manual Selection of multiple drivers (Example)

© Hilscher, 2009-2013

3.3.2 cifX Device Driver

In the DeviceNet Slave-DTM for the **cifX Device Driver** no driver dialog pane is available, since for the **cifX Device Driver** no driver settings are required.

The **cifX Device Driver** will be used if the DeviceNet Slave DTM is installed in the same PC as the DeviceNet Slave device.

Note: To establish a connection from a DTM to a Slave device via the **cifX Device Driver**, the **cifX Device Driver** must be installed and the driver must have access to the Slave device.

3.3.3 netX Driver

The **netX Driver** is used to connect the DTM to the device via different connection types. The DTM communicates with the device via an USB connection, a serial (RS232) connection or a TCP/IP connection. The **netX Driver** establishes

- via the USB interface of the device and the USB port of the PC an USB connection to the device,
- via the RS232 interface of the device and the COM port of the PC a serial connection (RS232) to the device
- and via Ethernet a TCP/IP connection to the device.

To connect the DTM to the physical layer of the device the **netX Driver** software works in combination with the software components:

- “USB/COM connector” for the USB connection and for the serial connection (RS232) and
- “TCP/IP connector” for the Ethernet connection.

© Hilscher, 2009-2013

3.3.4 Configuring netX Driver

The following steps are required to configure the netX Driver:

USB/RS232 Connection

To set the driver parameters for an USB/RS232 connection note:

Note: Adjust the driver parameters netX Driver USB/RS232 only if they differ from the default settings. After saving the changed driver parameters, these parameters are used for the device assignment when scanning devices.

For setting the driver parameters for an USB connection or a serial connection:

1. Select **Settings > Driver > netX Driver > USB/RS232 Connection**.

➢ Set the driver netX Driver USB/RS232 parameters.

TCP/IP Connection

For setting the driver parameters for a TCP/IP connection:

1. Select **Settings > Driver > netX Driver > TCP Connection**.
2. Set IP Address of the device:

➢ Add an IP Range via **Select IP Range** .

3. Under **IP Range Configuration** > **IP Address** enter the IP Address of the device (**Use IP Range** is unchecked).

Or

4. Set IP Range:

➢ Check **Use IP Range**.

➢ Under **IP Range Configuration** > **IP Address** enter the start address (left side) and the ending address of the IP scanning range (right side).

5. Click **Save**, to save the IP address or the IP range.

After saving the changed driver parameters, these parameters are used for the device assignment when scanning devices.

© Hilscher, 2009-2013

3.3.5 netX Driver - USB/RS232 Connection

The communication from the DTM to the device via an **USB/RS232 Connection** is used when the DTM is installed on a PC and between the PC and the device

- an USB connection
- or a serial connection (RS232) exists.

The DTM accesses the device via the USB interface or via the RS232 interface. This requires either to connect an USB port of the PC to the USB interface of the device using an USB cable or to connect a physical COM port of the PC to the RS232 interface of the device via a serial cable.

The **netX Driver** / **USB/RS232 Connection** supports all physical and virtual COM ports available on the PC.

Via the RS232 interface or USB interface, the device is configured or diagnosis is performed.

© Hilscher, 2009-2013

3.3.5.1 Driver Parameters for netX Driver - USB/RS232 Connection

The settings of the driver parameters for the USB/RS232 connection are made via the **netX Driver / USB/RS232 Connection** configuration dialog.

➢ Open the **USB/RS232 Connection** dialog via navigation area **Settings > Driver > netX Driver**.

➢ The **USB/RS232 Connection** dialog is displayed:

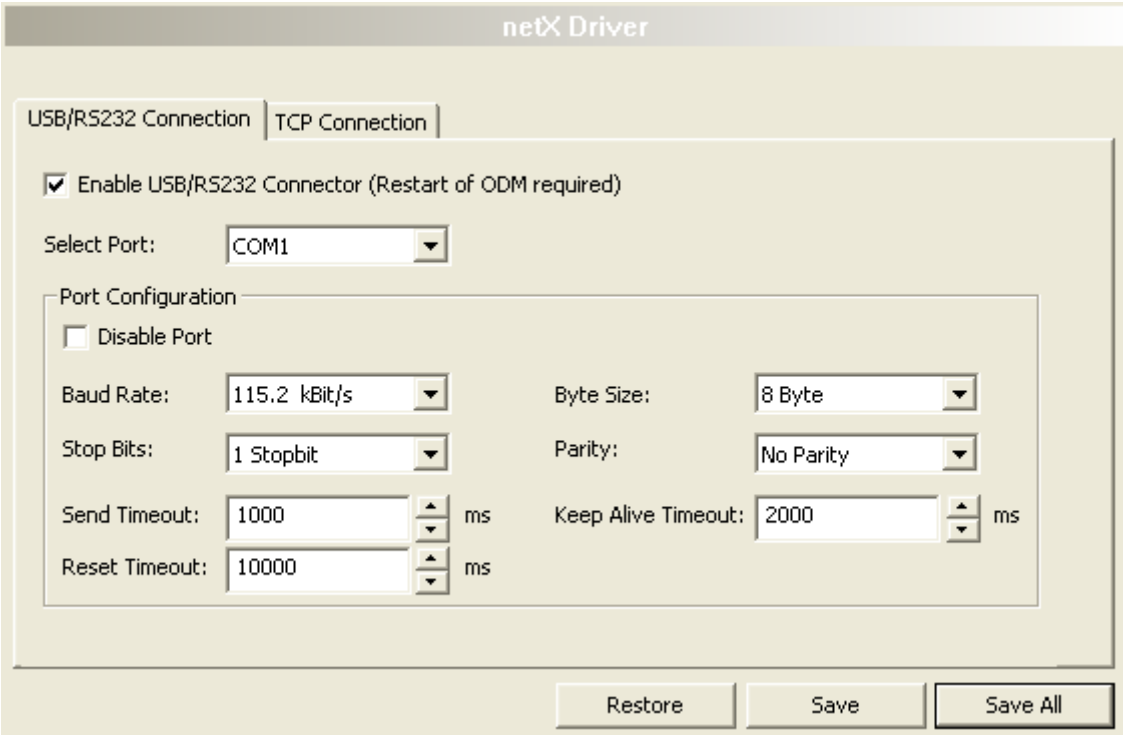


Figure 9: netX Driver > USB/RS232 Connection

Parameter	Meaning	Range of Value / Default Value
Enable USB/RS232 Connector (Restart of ODM required)	checked: The netX Driver can communicate via the USB/RS232 interface. unchecked: The netX Driver can not communicate via the USB/RS232 interface. If the check mark for **Enable USB/RS232 Connector** is set or removed, then the ODM server must be restarted[1], to make the new setting valid. [1] Restart the ODM server via the **ODMV3 Tray Application**: -In the foot line click on using the right mouse key. - In the context menu select **Service > Start**.	checked, unchecked; Default: unchecked
Select Port	Depending on the COM ports (interfaces) available on the PC, they will be listed under **Select Port**.	COM 1 to COM N
Port Configuration		
Disable Port	checked: No connection. unchecked: The netX Driver tries to establish a connection using the configured USB/RS232 interface.	checked, unchecked (Default)
Baud rate	Transfer rate: number of bits per second. The device must support the baud rate.	9.6, 19.2, 38.4, 57.6 or 115.2 [kBit/s]; Default (RS232): 115.2 [kBit/s]

© Hilscher, 2009-2013

Parameter	Meaning	Range of Value / Default Value
Stop bits	Number of stop bits sent after the transfer of the send data for synchronization purposes to the receiver.	Stop bit: 1, 1.5, 2; Default (RS232): 1
Send Timeout	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 … 60.000 [ms]; Default (RS232 and USB): 1000 ms
Reset Timeout	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 … 60.000 [ms]; Default (RS232 and USB): 5000 ms
Byte size	Number of bits per byte by byte specification	7 Bit, 8 Bit; Default (RS232): 8 Bit
Parity	In the error detection in data transmission using parity bits, "parity" describes the number of bits occupied with 1 in the transmitted information word. No Parity: no parity bit Odd Parity: The parity is "odd" if the number of bits occupied with 1 in the transmitted information word will be odd. Even parity: The parity is "even" if the number of bits occupied with 1 in the transmitted information word will be even. Mark Parity: if the parity bit is always 1, this will be named mark-parity (the bit does not contain any information). Space Parity: if the parity bit always 0, this will be named space-parity (the bit represents an empty space).	No Parity, Odd Parity, Even Parity, Mark Parity, Space Parity; Default (RS232): No Parity
Keep Alive Timeout	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism will be initiated after the set time has elapsed if the communication has failed.	100 … 60.000 [ms]; Default (RS232 and USB): 2000 ms
Restore	Resets all settings in the configuration dialog to the default values.	
Save	Saving all settings made in the configuration dialog **netX Driver > Save USB/RS232 Connection**, i. e. only for the selected connection type.	
Save All	Saving all settings made in the configuration dialog **netX Driver**, i. e. for all connection types.	

Table 11: Parameters netX Driver > USB/RS232 Connection

© Hilscher, 2009-2013

3.3.6 netX Driver - TCP/IP Connection

The communication from the DTM to the device via a **TCP/IP Connection** is used in the following two typical applications:

Application 1: The device has its own Ethernet interface. The DTM is installed on a PC and the TCP/IP connection is established from this PC to the stand-alone device. The IP address of the device is used.

Application 2: The device is installed in a remote PC. The DTM is installed on an additional PC and the TCP/IP connection is established from this PC to the remote PC. The IP address of the remote PC is used. For the TCP/IP connection is made, on the remote PC the cifX TCP/IP server must be started. The cifX TCP/IP server allows the remote access to the device via a TCP/IP connection.

Note: An exe file for the cifXTCP/IP server is provided on the product CD in the *Tools* directory.

Via the TCP/IP interface of the device or of the remote PC, the device is configured or diagnosis is performed.

© Hilscher, 2009-2013

3.3.6.1 Driver Parameters for netX Driver - TCP/IP Connection

The settings of the driver parameters for the TCP/IP connection are made via the **netX Driver / TCP Connection** configuration dialog.

➢ Open the **TCP Connection** dialog via navigation area **Settings > Driver > netX Driver**.

↬ The dialog **netX Driver** is displayed:

➢ Select **TCP Connection**.

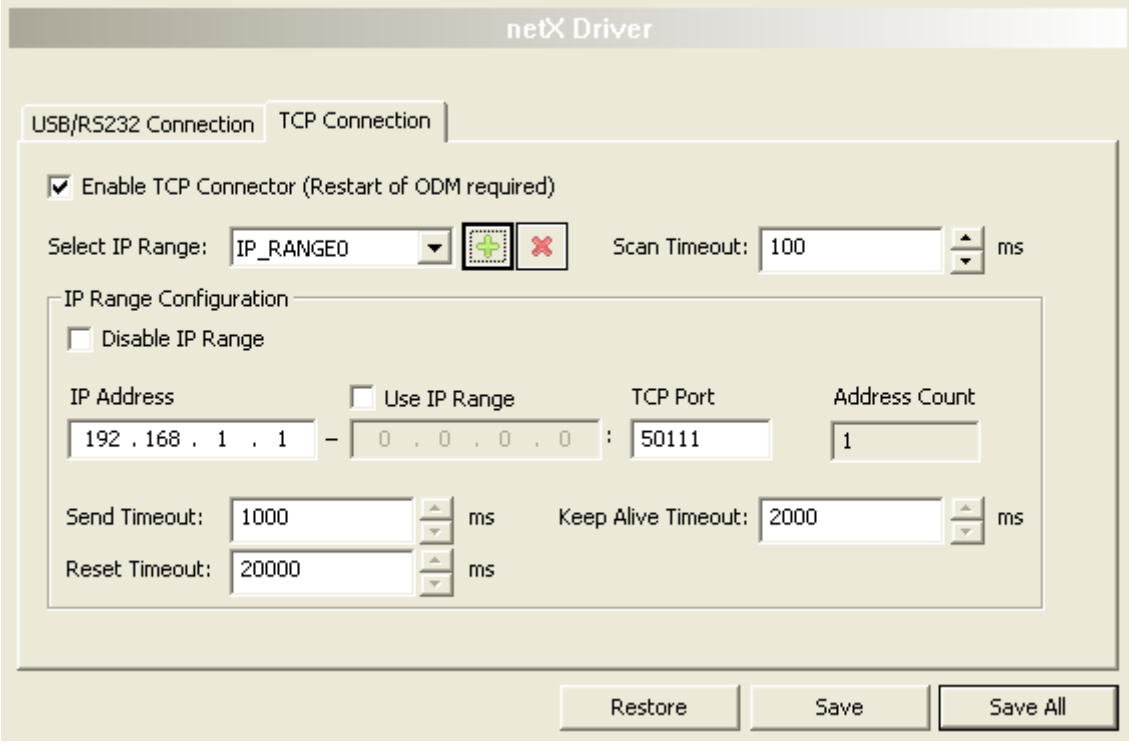


Figure 10: netX Driver > TCP Connection

Parameter	Meaning	Range of Value / Default Value
Enable TCP Connector (Restart of ODM required)	checked: The netX Driver can communicate via the TCP/IP interface. unchecked: The netX Driver can not communicate via the TCP/IP interface. If the check mark for **Enable TCP Connector** is set or removed, then the ODM server must be restarted[1], to make the new setting valid. [1] Restart the ODM server via the **ODMV3 Tray Application**: -In the foot line click on using the right mouse key. - In the context menu select **Service > Start**.	checked, unchecked; Default: unchecked
Select IP Range	Via **Select IP Range** already created IP ranges can be selected. Via an additional IP range can be added. Via an IP range can be deleted.	

© Hilscher, 2009-2013

Parameter	Meaning	Range of Value / Default Value
Scan Timeout [ms]	With **Scan Timeout** can be set, how long to wait for a response while a connection is established.	10 … 10000 [ms]; Default: 100 ms
IP Range Configuration		
Disable IP Range	checked: No connection. unchecked: The netX Driver tries to establish a connection using the configured TCP/IP interface.	checked, unchecked (Default)
IP Address (left)	Enter the IP address of the device, (if **Use IP Range** is not checked). Enter the start address of the IP scanning range, (if **Use IP Range** is checked).	valid IP address; Default: 192.168.1.1
Use IP Range	checked: An IP address range is used. unchecked: Only one IP address is used.	checked, unchecked; Default: unchecked
IP Address (right)	Enter the ending address of the IP scanning range, (only if **Use IP Range** is checked).	valid IP address; Default: 0.0.0.0
Address Count	Displays the scanning range address count, depending on the selected IP-start or IP-end address. (For this read the note given below.)	recommended: 10
TCP Port	Identifies the endpoint of a logical connection or addresses a specific endpoint on the device or PC.	0 - 65535; Default Hilscher device: 50111
Send Timeout [ms]	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 … 60.000 [ms]; Default (TCP/IP): 1000 ms
Reset Timeout [ms]	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 … 60.000 [ms]; Default (TCP/IP): 2000 ms
Keep Alive Timeout [ms]	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism will be initiated after the set time has elapsed if the communication has failed.	100 … 60.000 [ms]; Default (TCP/IP): 2000 ms
Restore	Resets all settings in the configuration dialog to the default values.	
Save	Saving all settings made in the configuration dialog **netX Driver > Save TCP/IP Connection**, i. e. only for the selected connection type.	
Save All	Saving all settings made in the configuration dialog **netX Driver**, i. e. for all connection types.	

Table 12: Parameters netX Driver > TCP Connection

Note: Do not use large IP ranges in combination with a low scan timeout. Microsoft introduced in Windows® XP SP2 a limit of concurrent half-open outbound TCP/IPconnections (connection attempts) to slow the spread of virus and malware from system to system. This limit makes it impossible to have more than 10 concurrent half-open outbound connections. Every further connection attempt is put in a queue and forced to wait. Due to this limitation a large IP range used in combination with a low scan timeout could prevent the connection establishment to a device.

© Hilscher, 2009-2013

3.4 Device Assignment

Note: In the **Device Assignment** dialog pane you first must assign the DeviceNet Slave device to the DeviceNet Slave DTM by checking the check box. This is essential to establish an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device later, as described in section *Connecting/Disconnecting Device* on page 66.

Therefore in the **Device Assignment** dialog pane you scan for the DeviceNet Slave device and select it.

If the device did not get a firmware or shall get a new firmware:

1. first you scan for the device (with or without firmware) and select the device,
2. then you download a firmware to the device and
3. subsequently you scan for the device (with firmware) once more and select the device again.

Important: For a 2-channel device consecutively channel 1 or channel 2 must be assigned to the DTM.

3.4.1 Scanning for Devices

1. Select **Settings > Device Assignment** in the navigation area.

↬ The dialog pane **Device Assignment** is displayed.

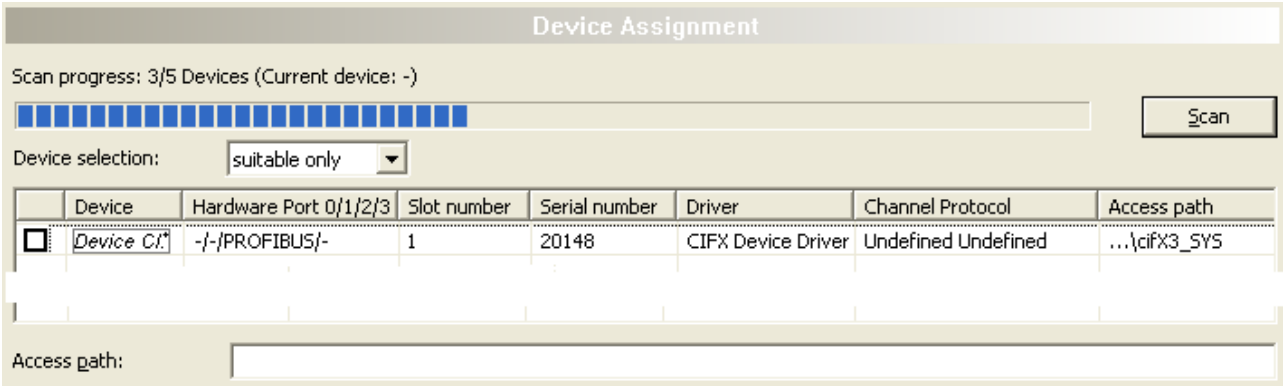


Figure 11: Device Assignment - detected Devices (The name of the device class is displayed.) – Example for a device without firmware*

2. Under **Device Selection** select *suitable only*.
3. Select **Scan**, to start the scanning process.

↬ In the table all devices are displayed, which can be connected to the DeviceNet Slave DTM via the preselected driver.

Note: For devices, which have been found via the **cifX Device Driver** in the column **Access path** the indication …\cifX[*0toN*]**_SYS** is displayed. This is correct, as long as a device did not get a firmware. After the firmware download has been completed, in the column **Access path** the indication …\cifX[*0toN*]**_Ch**[*0to3*] is displayed.

© Hilscher, 2009-2013

Parameter	Meaning	Range of Value / Default Value
Device selection	Selecting **suitable only** or **all** devices.	suitable only, all
Device	Device class of the DeviceNet Slave devices.	
Hardware Port 0/1/2/3	Shows, which hardware is assigned to which communication interface.	
Slot number	Shows the **Slot Number (Card ID)** preset at the PC card cifX via the **Rotary Switch Slot Number (Card ID)**. The indication **n/a** means that no **Slot-Nummer (Card ID)** exists. This will occur if the PC card cifX is not equipped with a **Rotary Switch Slot Number (Card ID)** or for PC cards cifX equipped with a **Rotary Switch Slot Number (Card ID)** if the rotary switch is set to the value 0 (zero).	1 to 9, n/a
Serial number	Serial number of the device	
Driver	Name of the driver	
Channel Protocol	Shows, which firmware is loaded to which device channel. The data for the used channel consists of the protocol class and the communication class. a.) For devices without firmware: Undefined Undefined, b.) For devices with firmware: Protocol name corresponding to the used Firmware	
Access path (last column on the right)	Depending on the used driver in the column **Access path** different data to the device are displayed. For the **cifX Device Driver** the following data are displayed: a.) For devices without firmware: …\cifX[*0toN*]**_SYS**, b.) For devices with firmware: …\cifX[*0toN*]_**Ch**[*0to3*]. cifX[*0toN*] = Board number 0 to N Ch[*0to3*] = Channel number 0 to 3	Depending on the device and on the driver: board or channel number, IP address or COM interface
Access path (at the lower side of the dialog pane)	If in the table a device is checked, under **Access path** (at the lower side of the dialog pane) the driver identification or depending on the used driver additional data to the device will be displayed. For the **cifX Device Driver** the following data are displayed: a.) For devices without firmware: …\cifX[*0toN*]**_SYS**, b.) For devices with firmware: …\cifX[*0toN*]_**Ch**[*0to3*]. cifX[*0toN*] = Board number 0 to N Ch[*0to3*] = Channel number 0 to 3	driver identification (ID) depending on the device and on the driver: board or channel number, IP address or COM interface

Table 13: Parameters of the Device Assignment

© Hilscher, 2009-2013

3.4.1.1 Scanning for all Devices or for suitable only

all

1. Under **Device Selection** select *all*.
2. Select **Scan**.

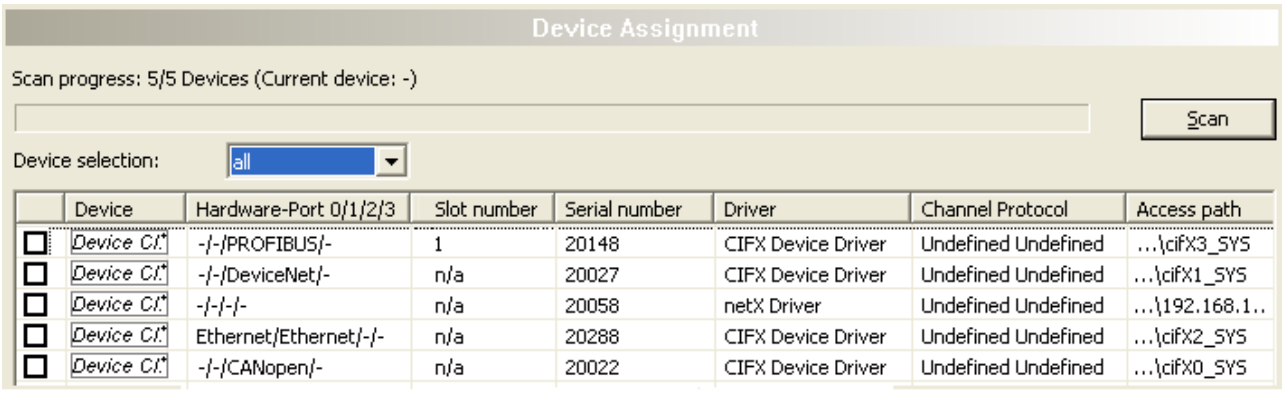


Figure 12: Device Assignment - detected Devices (The name of the device class is displayed.) Example for Devices without Firmware*

➮ In the table all devices are displayed, which are attainable in the network and which can be connected to a single DTM each via the preselected drivers.

Note: During a subsequent firmware download in the selection window **Select Firmware File** all files from the selected folder are displayed, under **Files of Type** „*All Files (*.*)*“ is displayed and the check box **Validate the selected firmware file.** is unchecked.

suitable only

1. Under **Device Selection** select *suitable only*.
2. Select **Scan**.

➮ In the table all devices are displayed, which can be connected to the DeviceNet Slave DTM via the preselected drivers.

Note: During a subsequent firmware download in the selection window **Select Firmware File** only firmware files from the selected folder are displayed, under **Files of Type** „*Firmware Files (*.nxm)*“ or „*Firmware Files (*.nxf)*“ is displayed and the check box **Validate the selected firmware file.** is checked.

© Hilscher, 2009-2013

3.4.2 Selecting the Device (with or without firmware)

Note: A connection with the DeviceNet Slave DTM can only be established with one DeviceNet Slave device.

To select the physical DeviceNet Slave device (with or without firmware):

1. Check the appropriate device.

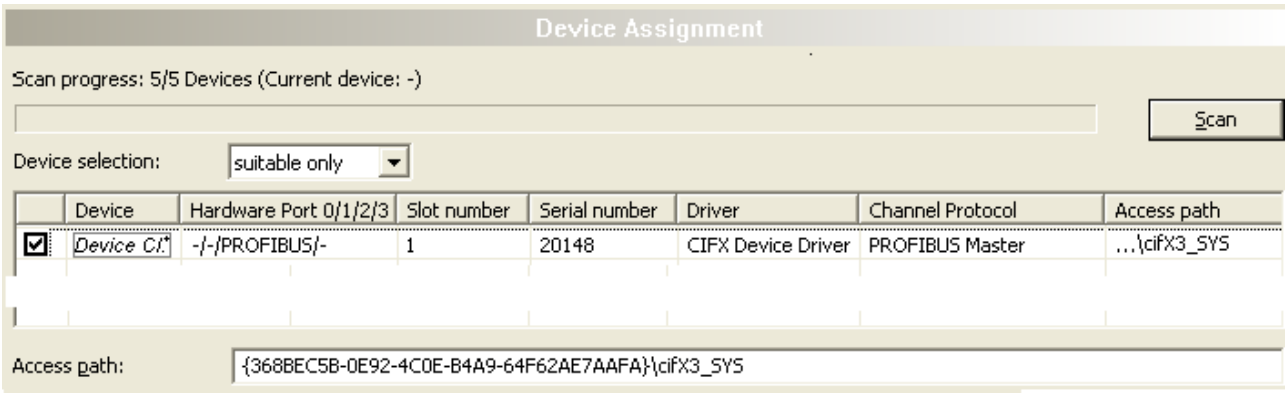


Figure 13: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for a device without firmware / one Device is selected*

- Under **Access path** (below in the dialog pane) the access path to the device, e. g. the driver identification, or depending on the used driver additional access data of the device are displayed.

2. Select **Apply**, to apply the selection.

Note: Before an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device can be established, a firmware must be loaded to the device and the device must be selected once more.

For further information refer to section to section *Firmware Download* on page 43 or to section *Selecting the Device once more (with Firmware)* on page 41.

© Hilscher, 2009-2013

3.4.3 Selecting the Device once more (with Firmware)

Note: For repeated download this step is omitted.

To select the DeviceNet Slave device (with firmware and defined system channel) once more, proceed as described hereafter:

all

1. Under **Device Selection** select *all*.
2. Select **Scan**.

   In the table all devices are displayed, which are attainable in the network and which can be connected to a DTM via the preselected drivers.
3. Check the appropriate device.

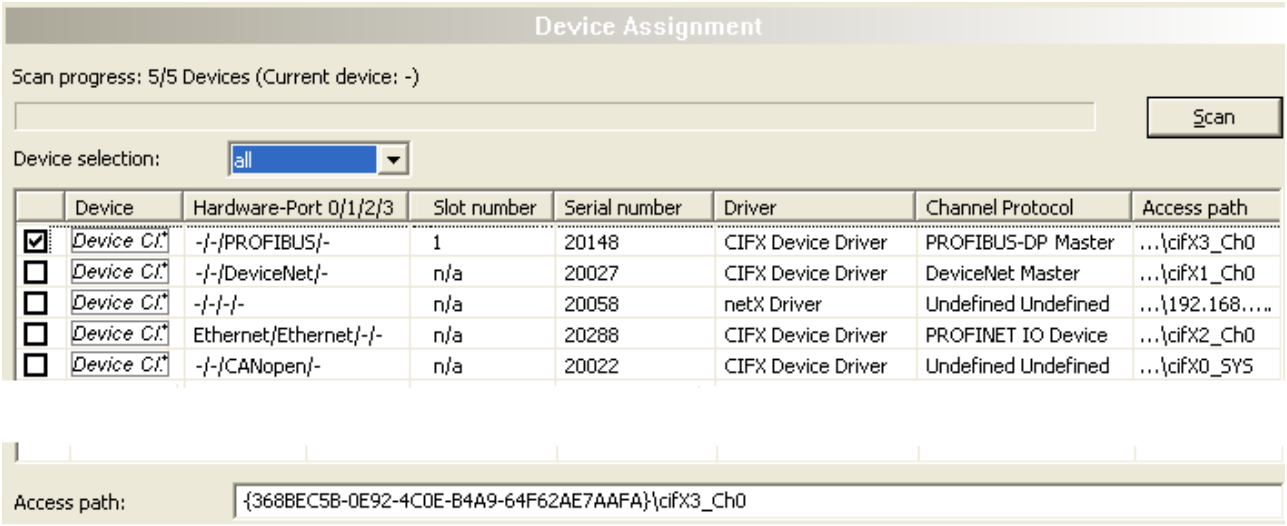


Figure 14: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for Devices with and without Firmware / one Device is selected*

Note: After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog pane): the data: …\cifX[*0toN*]_Ch[*0to3*].

  cifX[*0toN*] = board number 0 to N
  Ch[*0to3*] = channel number 0 to 3

4. Select **Apply**, to apply the selection.
5. Or select **OK**, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).

© Hilscher, 2009-2013

Or:

suitable only

1. Under **Device Selection** select *suitable only*.
2. Select **Scan**.

↬ In the table all devices are displayed, which can be connected to the DeviceNet Slave DTM via the preselected drivers.

3. Check the appropriate device.

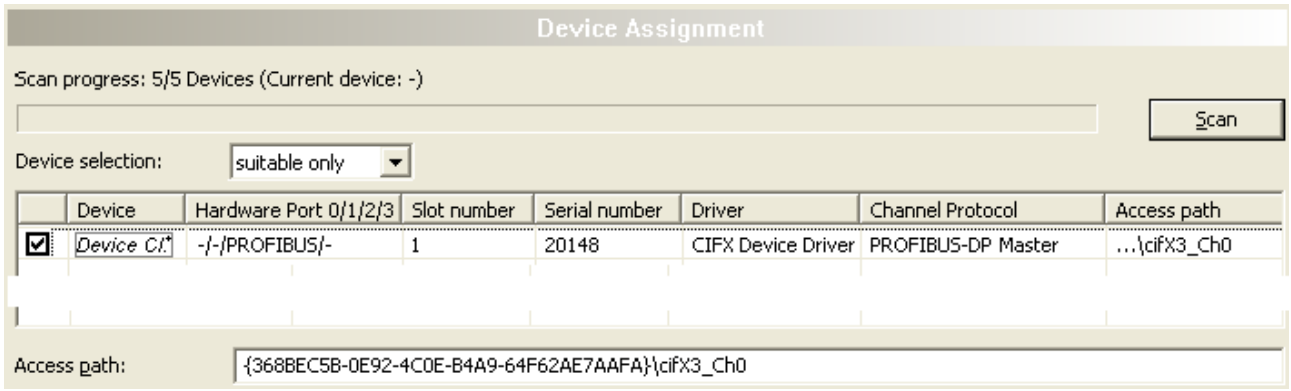


Figure 15: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for a device with firmware / one Device is selected*

Note: After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog pane): the data: …\cifX[*0toN*]_Ch[*0to3*].

  cifX[*0toN*] = board number 0 to N
  Ch[*0to3*] = channel number 0 to 3

4. Select **Apply**, to apply the selection.
5. Or select **OK**, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).

For further information how to establish an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device, refer to section *Connecting/Disconnecting Device* on page 66.

© Hilscher, 2009-2013

3.5 Firmware Download

Using the **Firmware Download** dialog a firmware can be transferred to the device.

CAUTION! The firmware download
- Stops the bus communication,
- erases the configuration data base and
- overwrites the existing firmware in the device.

To complete the update and to make the device operable again, please re-download the configuration when this operation has finished.

Note: Prior to the firmware download, you must select the driver and the Slave device (with or without firmware).

For further information refer to section *Overview Settings* on page 24.

To load the firmware to the device:

1. In the navigation area select **Settings > Firmware Download**.

➪ The dialog **Firmware-Download** pane is displayed.

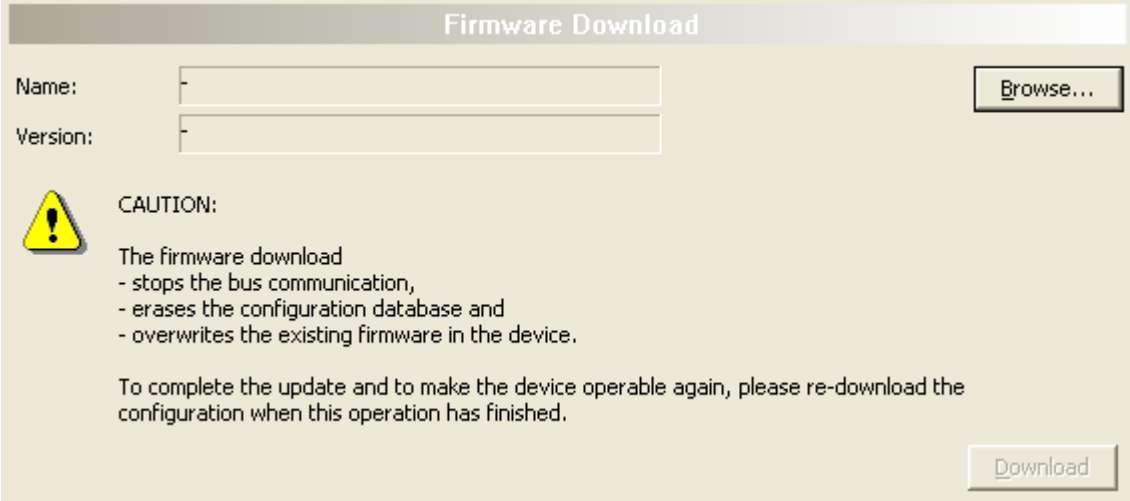

Firmware Download
Name:
Version:
Browse...
CAUTION:
The firmware download
- stops the bus communication,
- erases the configuration database and
- overwrites the existing firmware in the device.
To complete the update and to make the device operable again, please re-download the configuration when this operation has finished.
Download

Figure 16: Firmware Download

Parameter	Meaning
Name	Name of the Firmware
Version	Version and Build Version of the Firmware
Browse…	Button to select the firmware file.
Download	Button to download the firmware.

Table 14: Parameter Firmware Download

2. Select **Browse**.

➪ The selection window **Select Firmware File** is displayed.

➪ Enlarge the selection window to view the columns **Hardware** and **Version**.

© Hilscher, 2009-2013

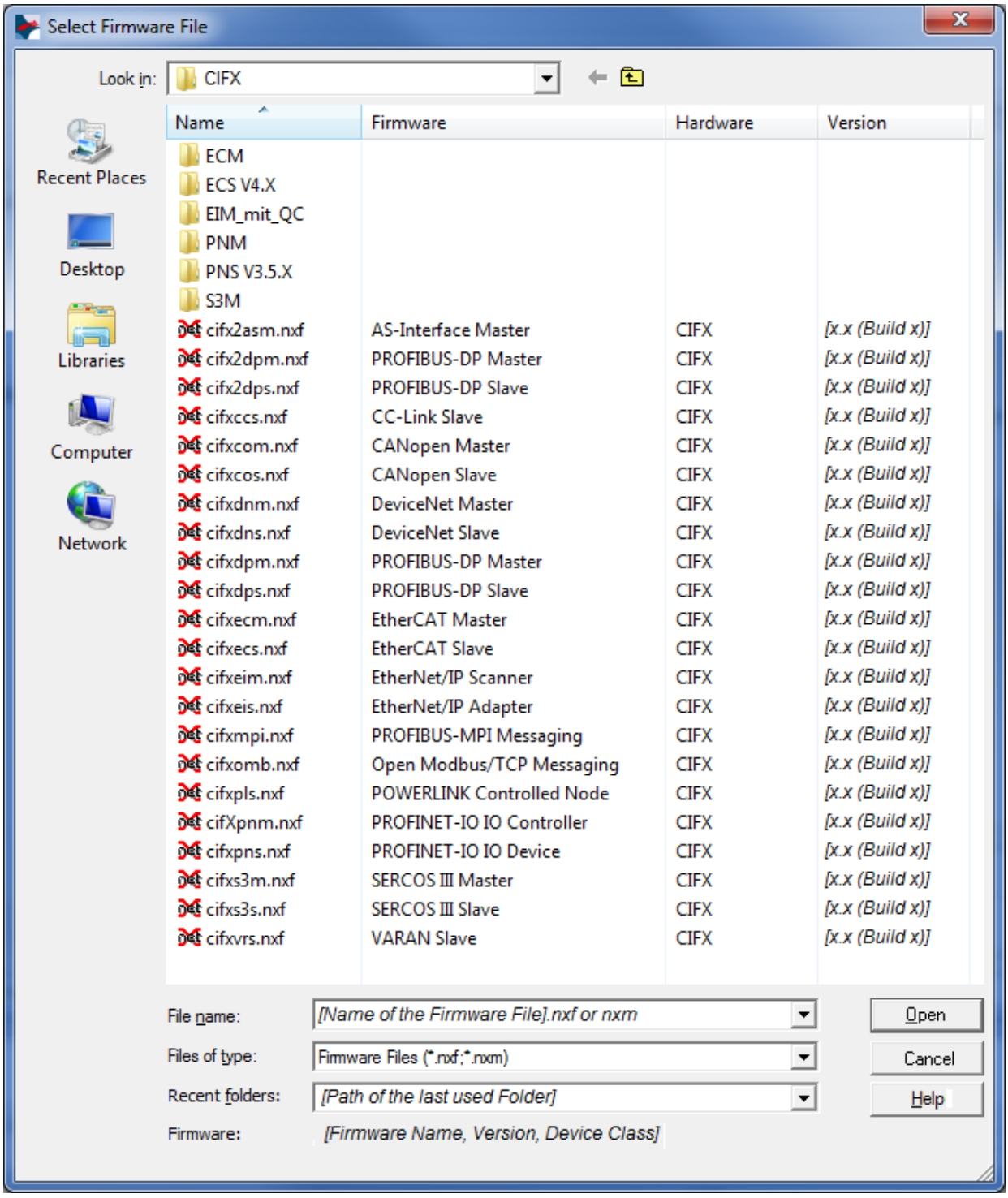

Select Firmware File
Look in: CIFX
Recent Places
Desktop
Libraries
Computer
Network
Name
Firmware
Hardware
Version
ECM
ECS V4.X
EIM_mit_QC
PNM
PNS V3.5.X
S3M
cifx2asm.nxf AS-Interface Master CIFX [x.x (Build x)]
cifx2dpm.nxf PROFIBUS-DP Master CIFX [x.x (Build x)]
cifx2dps.nxf PROFIBUS-DP Slave CIFX [x.x (Build x)]
cifxccs.nxf CC-Link Slave CIFX [x.x (Build x)]
cifxcom.nxf CANopen Master CIFX [x.x (Build x)]
cifxcos.nxf CANopen Slave CIFX [x.x (Build x)]
cifxdnm.nxf DeviceNet Master CIFX [x.x (Build x)]
cifxdns.nxf DeviceNet Slave CIFX [x.x (Build x)]
cifxdpm.nxf PROFIBUS-DP Master CIFX [x.x (Build x)]
cifxdps.nxf PROFIBUS-DP Slave CIFX [x.x (Build x)]
cifxecm.nxf EtherCAT Master CIFX [x.x (Build x)]
cifxecs.nxf EtherCAT Slave CIFX [x.x (Build x)]
cifxeim.nxf EtherNet/IP Scanner CIFX [x.x (Build x)]
cifxeis.nxf EtherNet/IP Adapter CIFX [x.x (Build x)]
cifxmpi.nxf PROFIBUS-MPI Messaging CIFX [x.x (Build x)]
cifxomb.nxf Open Modbus/TCP Messaging CIFX [x.x (Build x)]
cifxpls.nxf POWERLINK Controlled Node CIFX [x.x (Build x)]
cifXpnm.nxf PROFINET-IO IO Controller CIFX [x.x (Build x)]
cifxpns.nxf PROFINET-IO IO Device CIFX [x.x (Build x)]
cifxs3m.nxf SERCOS III Master CIFX [x.x (Build x)]
cifxs3s.nxf SERCOS III Slave CIFX [x.x (Build x)]
cifxvrs.nxf VARAN Slave CIFX [x.x (Build x)]
File name: [Name of the Firmware File].nxf or nxm
Files of type: Firmware Files (*.nxf;*.nxm)
Recent folders: [Path of the last used Folder]
Firmware: [Firmware Name, Version, Device Class]
Open
Cancel
Help

Figure 17: Window 'Select Firmware File' (Example CIFX)

Parameter	Meaning	Range of Value / Default Value
Column **Name**	File name of the firmware file To sort the entries of the window **Select Firmware File** by name click to the column head Name.	nxf, nxm
Column **Firmware**	Name of the firmware (consisting of the protocol name and protocol class)	
Column **Hardware**	Device class of the associated hardware	e. g. CIFX, COMX, COMX 51, NETJACK 10, NETJACK 50,

© Hilscher, 2009-2013

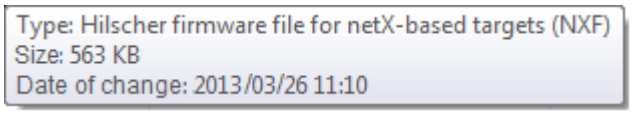
		NETJACK 51, NETJACK 100, NETTAP 50 (Gateway), NETTAP 100 (Gateway), NETBRICK 100 (Gateway)
Column **Version**	Firmware version	x.x (build x)
Tooltip	To view the tooltip information move with the mouse pointer over the selected firmware line. 	
Files of Type	„*All Files (*.*)*“ if before in the **Device Assignment** pane under **Device selection** *all* was selected. „ *Firmware Files (*.nxm)*“ or *Firmware Files (*.nxf)* if before in the **Device Assignment** pane under **Device selection** *suitable only* was selected.	*All Files (*.*)*, *Firmware Files (*.nxm)*, *Firmware Files (*.nxf)*
Recent folders	Path of the recently opened folder	
Firmware	As soon as the firmware file has been selected, under **Firmware** the name, the version and the build version as well as the device class for the selected firmware is displayed.	Name, Version, Build Version, Device Class for the selected firmware
Help	Button, to open the online help of the DTM.	

Table 15: Parameters Select Firmware File

Further descriptions to the selection window **Select Firmware File** are included in the context sensitive help (**F1** key) of the Microsoft Corporation.

Note: After in the **Device Assignment** pane under **Device selection** *all* or *suitable only* has been set, during a subsequent firmware download in the selection window **Select Firmware File** the following data are displayed or set:

(for list box entry →)	**all**	**suitable only**
In the selection window **Select Firmware File**:	all files from the selected folder	only firmware files from the selected folder
Under **Files of Type***:	„*All Files (*.*)*“	„*Firmware Files (*.nxm)*“, „*Firmware Files (*.nxf)*“
Validation:	A restricted validation will be performed if the selected firmware is applied for the download.	A validation is made whether the firmware file is suitable for the DeviceNet Slave DTM.

*These settings in the selection window **Select Firmware File** can also be changed manually.

3. In the selection window mark the firmware file to be loaded using the mouse.

In the selection window under **Firmware** the name and the version of the firmware are displayed.

4. In the selection window select the **Open** button.

A validation is made, whether the selected firmware file is suitable for the DeviceNet Slave DTM.

In the dialog pane **Firmware Download** during the validation **Download** is grayed out.

The selection window is closed.

5. In the dialog pane **Firmware Download** click to the **Download** button,

© Hilscher, 2009-2013

to download the firmware.

Firmware is not valid for the selected device

If a firmware file is selected, which is not valid for the assigned device, the request **Select Firmware File** will be displayed.

Figure 18: Request Select Firmware File - Example

➢ Answer to the request with **No**.

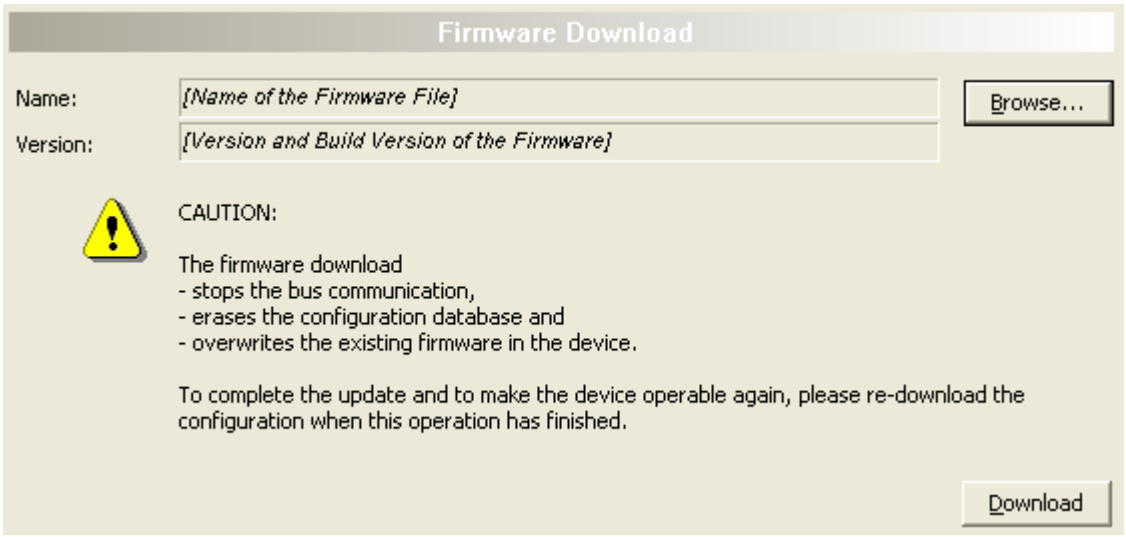

Firmware Download
Name:
[Name of the Firmware File]
Browse...
Version:
[Version and Build Version of the Firmware]
CAUTION:
The firmware download
- stops the bus communication,
- erases the configuration database and
- overwrites the existing firmware in the device.
To complete the update and to make the device operable again, please re-download the configuration when this operation has finished.
Download

Figure 19: Firmware Download – Download

The request **Do you really want to download the firmware?** is displayed.

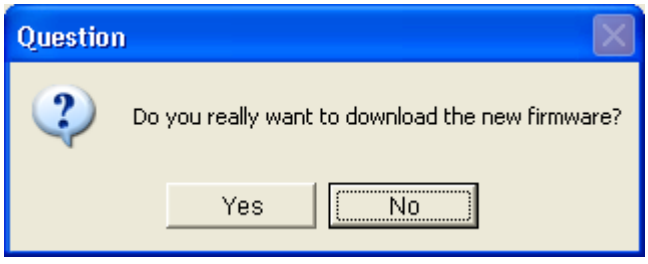


Figure 20: Firmware-Download –Question

© Hilscher, 2009-2013

During the download a progress bar is displayed, in the status line a clock / green hook symbol is displayed and **Download** is grayed out.

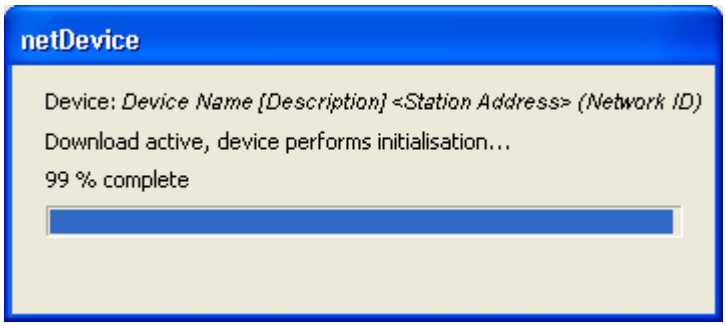


Figure 21: Clock Symbol and Hook Symbol green

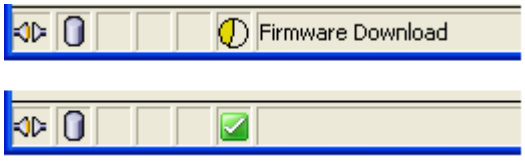


Figure 22: Firmware Download - Progress Bar

© Hilscher, 2009-2013

4 Configuration

4.1 Overview Configuration

Configuration Dialog Panes

The table below gives an overview for the **Configuration** dialog panes descriptions:

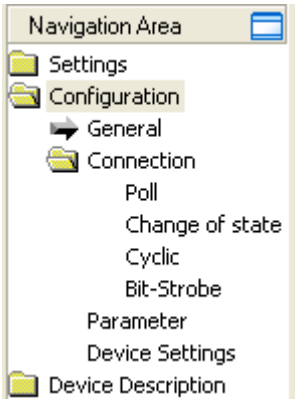
DeviceNet Slave-DTM	Folder Name / Section	Page
 *Navigation Area - Configuration*	*General Settings*	*51*
	Connection Configuration	*53*
	Poll Connection	*54*
	Change of State Connection	*56*
	Cyclic Connection	*58*
	Bit-Strobe Connection	*59*
	Parameter	*61*
	Device Settings	*63*
	Start of Bus Communication	
	Application Monitoring	*63*

Table 16: Dialog Panes Configuration

Notice the descriptions in the section *Configuration Steps* on page 18.

Note: In order to transfer the configuration to the DeviceNet Slave device, download the data of the configuration parameters in the DeviceNet Slave device. See section *Download Configuration* on page 67.

© Hilscher, 2009-2013

4.2 Configuring Slave Parameters

Important: For a 2-channel device consecutively channel 1 or channel 2 each must be configured individually.

The steps provided in the following two sections are alternatively required for the two cases listed hereafter, to configure the parameters of the DeviceNet Slave device using the DeviceNet Slave DTM:

- Slave DTM at the Root-Line (Stand-Alone Slave)
- and Slave DTM at the Master busline.

4.2.1 Slave DTM at the Root-Line (Stand-Alone Slave)

If in the network project the DeviceNet Slave DTM is dropped to the root busline, proceed as follows:

General

1. Set UCMM and Fragmentation Timeout.

➢ Select **Configuration > General** in the navigation area.

Connection

2. Configure the device connection.

➢ Select **Configuration > Connection** in the navigation area.

Parameter

3. Set the parameter data of the device.

➢ Select **Configuration > Parameter** in the navigation area.

Device Settings

4. Set the Device Settings:

➢ Select **Configuration > Device Settings** in the navigation area.

➢ Under **Start of Bus Communication** select **Automatically by device** or **Controlled by application** option.

➢ Under **Application Monitoring** > set **Watchdog time**.

Close Slave DTM Configuration Dialog

5. Click **OK** in order to close the Slave DTM configuration dialog and to store your configuration.

Configuration Download to the DeviceNet Slave Device

Note: In order to transfer the configuration to the DeviceNet Slave device, download the data of the configuration parameters in the DeviceNet Slave device. See section *Download Configuration* on page 67.

© Hilscher, 2009-2013

Further Information

For more information refer to section *General Settings* on page 51, *Connection Configuration* on page 53, *Parameter* on page 61 and to section *Device Settings* on page *62* of this document.

4.2.2 Slave DTM at the Master busline

If in the network project the DeviceNet Slave DTM is dropped to the Master busline of the DeviceNet Master DTM:

General

1. Set UCMM and Fragmentation Timeout.

➢ Select **Configuration > General** in the navigation area.

Connection

2. Configure the device connection.

➢ Select **Configuration > Connection** in the navigation area.

Parameter

3. Set the parameter data of the device.

➢ Select **Configuration > Parameter** in the navigation area.

Device Settings

4. Set the Device Settings:

➢ Select **Configuration > Device Settings** in the navigation area.

➢ Under **Start of Bus Communication** select **Automatically by device** or **Controlled by application** option.

➢ Under **Application Monitoring** > set **Watchdog time**.

Close Slave DTM Configuration Dialog

5. Click **OK** in order to close the Slave DTM configuration dialog and to store your configuration.

Configuration Download to the DeviceNet Slave Device

Note: In order to transfer the configuration to the DeviceNet Slave device, download the data of the configuration parameters in the DeviceNet Slave device. See section *Download Configuration* on page 67.

Further Information

For more information refer to section *General Settings* on page 51, *Connection Configuration* on page 53, *Parameter* on page 61 and to section *Device Settings* on page *62* of this document.

© Hilscher, 2009-2013

4.3 General Settings

The Dialog **General Settings** contains the following configuration possibilities:

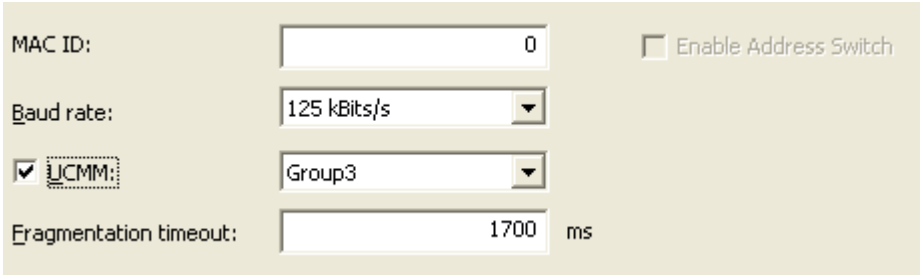


Figure 23: General Settings - Attributes of the device identification (Slave DTM at the Root-Line (Stand-Alone Slave))

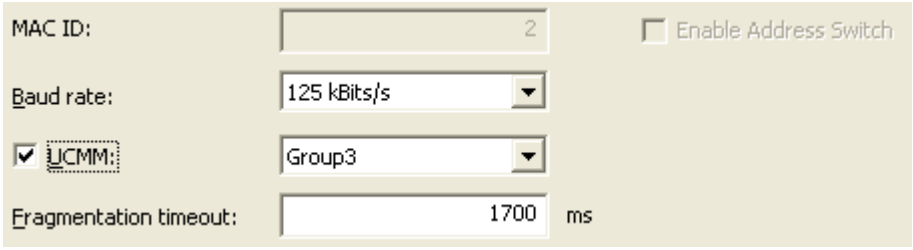


Figure 24: General Settings - Attributes of the device identification (Slave DTM at the Master busline)

Parameter	Meaning	Range of Value
MAC ID	The **MAC ID** is assigned by the Master and can not be edited here. Changing the MAC ID has to be made with the DeviceNet Master DTM. With each device inserted into the configuration the **MAC ID** is increased automatically by one. Slave DTM at the Root-Line (Stand-Alone Slave): The **MAC ID** is set in the Slave DTM. Slave DTM at the Master busline): The **MAC ID** is set in the Master DTM.	0 … 63
Baudrate	Baud rate of DeviceNet connection	500 kBaud, 250 kBaud, 125 kBaud, Default 500 kBaud
UCMM	If the field **UCMM** is selected, the device is used as UCMM-compatible device. The option **UCMM** is used for devices which need the UCMM message format. Group 1, 2 and 3 are supported. The documentation of the used device gives information whether this option is to be used or not.	Group1, Group2, **Group3**
Fragmentation Timeout	**Fragmentation Timeout** (Expl. Message Timeout): If an IO data transmission or an Explicit Message is larger than 8 byte, this must be transferred fragmented in the DeviceNet (in several telegrams). The **Fragmentation Timeout** specifies, how long the Master waits, until a Slave answers a fragmented telegram. **Note:** Small values can lead to communication disturbances.	0 … **1700** … 65535
Enable Address Switch	Refer to section *Enable Address Switch* on page 52 of this document.	

Table 17: General Settings - Attributes of the device identification

© Hilscher, 2009-2013

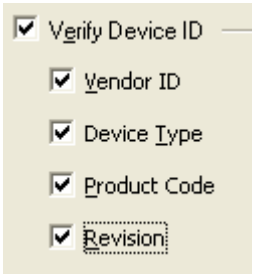
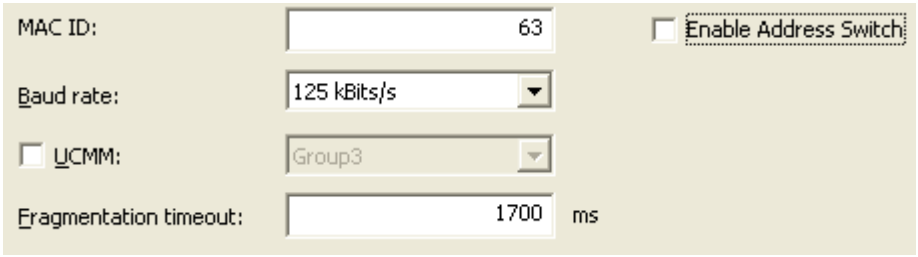


Figure 25: General Settings - Verify Device ID

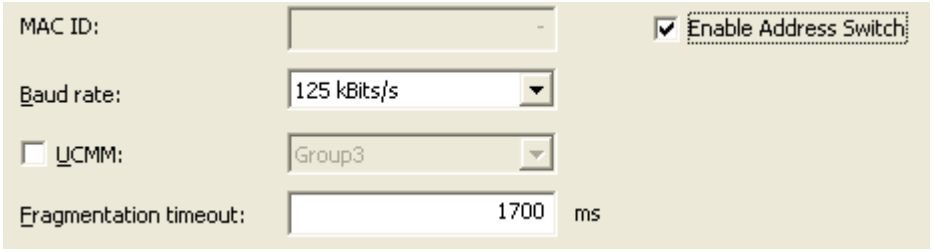
The function **Verify Device ID** compares the device description in the EDS file of the device with the existing hardware, if the device characteristics of the EDS file agree with those of the hardware.

The check is made for the selected attributes in each case.

4.3.1 Enable Address Switch



'Enable Address Switch' disabled



'Enable Address Switch' enabled

Figure 26: General Settings - Enable Address Switch (Slave DTM at the Root-Line (Stand-Alone Slave, Exmaple COMX 10XX-DNS/DNS))

Parameter	Meaning	Range of Value
Enable Address Switch	Defines, if the MAC ID is configured in the configuration software or at the address switch. If checked, the MAC ID is configured at the address switch. The parameter 'Enable Address Switch' with the setting 'checked' can only be used for COMX 10XX-DNS/DNS.	Default: CIFX DN/DNS, COMX 100XX-DN/DNS, NJ 100XX-DN/DNS: unchecked COMX 10XX-DNS/DNS: checked.

Table 18: General Settings - Enable Address Switch

© Hilscher, 2009-2013

4.4 Connection Configuration

DeviceNet allows establishing several kinds of **Connections** between devices. In DeviceNet a device (Slave) is mapped as a collection of objects. These objects communicate via different connection types, which you can adjust under **Connection**.

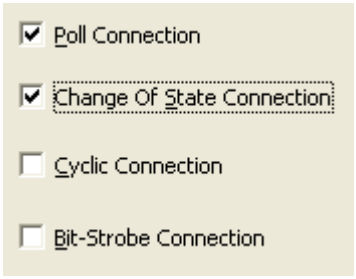


Figure 27: Configuration Dialog Connection

In the **Connection** dialog a connection type or a combination of types can be selected. Please note that a device has not to support all types of IO connections. Connection types which are not supported by the device are automatically disabled.

In the lower section of this dialog the possible combinations of the connection types are displayed:

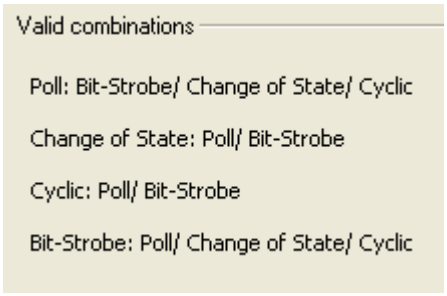

Valid combinations
Poll: Bit-Strobe/ Change of State/ Cyclic
Change of State: Poll/ Bit-Strobe
Cyclic: Poll/ Bit-Strobe
Bit-Strobe: Poll/ Change of State/ Cyclic

Figure 28: Indication of possible combinations of connection types

If an invalid combination is set, the following warning appears:

Figure 29: Warning invalid connection type combination

4.4.1 Connection Types

The following connection types are available:

Connection Type	Page
Poll Connection	54
Change of State Connection	56
Cyclic Connection	58
Bit-Strobe Connection	59

Table 19: Possible connection types

© Hilscher, 2009-2013

4.4.1.1 Poll Connection

If the **Poll Connection** type was activated, the elements of this connection type are editable. Otherwise this dialog is disabled.

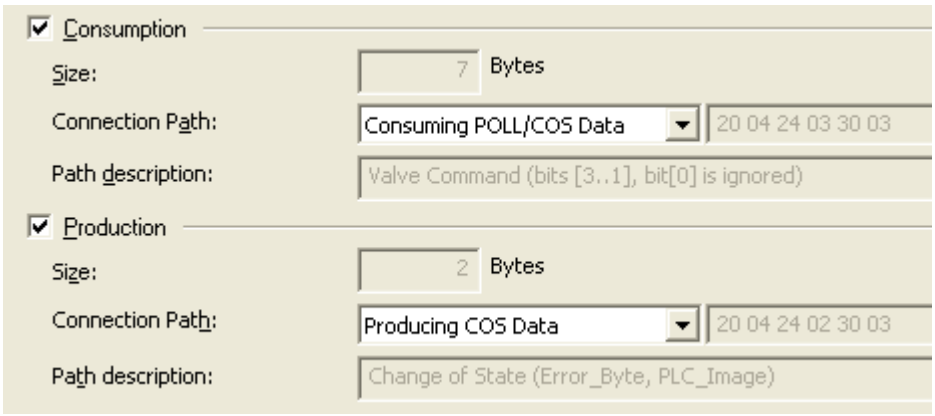


Figure 30: Poll Connection - Consumption and Production

One poll command from the Master sends a number of output data in the poll command to the device. The device receives (consumes) the output data. If it has input data configured for this poll connection it reacts by sending (producing) back the number of input data to the Master. Before a polled I/O connection is initiated by the Master, it reads the consumed and produced connection size of the data from the device (Slave) first and compares this values with the values configured in Master. If different values are detected, the connection cannot be established. A poll command can be sent from the Master to a device. The device has to respond if it has received the poll command of the Master, even if it has no input data. Else the Master will report a timeout error. Polling data to many devices has the disadvantage that the network traffic rate is very high and most data which is transferred has not changed since the last transmission. Furthermore the higher the bus load more communication errors can occur if the bus is disturbed by external influences.

Figure 31: Poll Connection – Timing

© Hilscher, 2009-2013

The **Production Inhibit Time**, one for each connection, configures the minimum delay time between new data production in multiples of a millisecond. The timer is reloaded each time new data production through the established connection occurs. While the timer is running the device suppresses new data production until the timer has expired. This method prevents that the device is overloaded with to fast incoming requests.

The value 0 defines no **Production Inhibit Time** and data production can and will be done as fast as possible. If in polled mode for example a Production Inhibit Time of 1000 ms is configured, then the poll request message to the device will be sent every second.

The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer. The fourfold value is used by the device later to reload its 'Watchdog Timer'. If no data production of the remote station takes place within this time, so the connection changes into a watchdog timeout error. Incoming data productions of the remote station load the Watchdog Time again to the fourfold value of the **Expected Packet Rate**.

Note: the **Production Inhibit Time** is verified against the **Expected Packet Rate**. If the **Expected Packet Rate** value is unequal to zero, but less than the **Production Inhibit Time** value, then an error message is displayed by the application.

The **Watchdog Timeout Action** defines the device behavior when the watchdog timer in the device (Slave) expires. The following actions are adjustable:

- **Timeout:** The connection transitions to the timeout state and remains in this state until it is Reset or Deleted.
- **Auto reset:** The connection remains in the established state and immediately restarts the Inactivity/Watchdog timer.
- **Auto delete:** The connection class automatically deletes the connection if it experiences an Inactivity/Watchdog timeout.

To reset the defalut values from the EDS file, click

.

© Hilscher, 2009-2013

4.4.1.2 Change of State Connection

If the **Change of State Connection** type was activated, the elements of this connection type are editable. Otherwise this dialog is disabled.

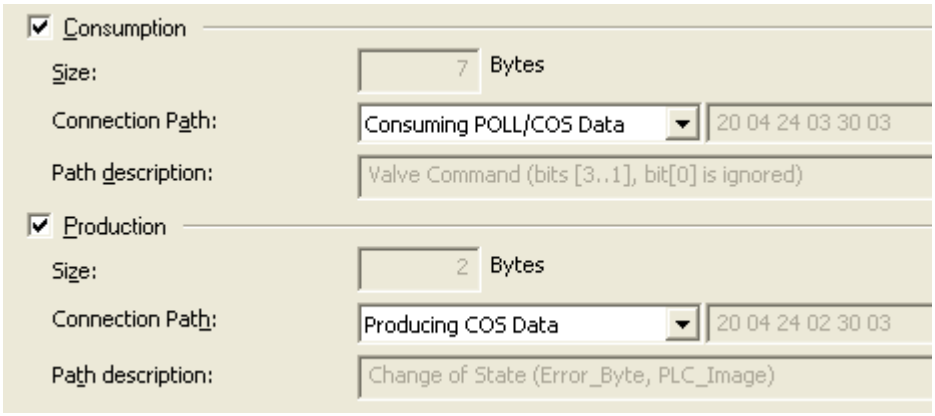


Figure 32: Change of State Connection - Consumption and Production

With this type of connection both Master and Slave send the configured amount of data (max. 255 Byte) to the respective remote station. This data production is started at change in value (trigger). If the data production does not take place during a defined time interval, the devices trigger the data production automatically to load the Watchdog Timer of the connection again. Depending on how the device behavior is configured, they can send back a confirmation message which contains any quantity of data and/or status information. Before a **Change of State** connection is initialized by the Master, it reads out the consumed and produced connection size of the data from the device (Slave) and compares this values with the values configured in the Master during configuration. If different values are determined, the connection can not build up. Data production only over 'Change of State' keeps the bus load as low as possible, while data than can be transmitted as fast as possible by each device because bus conflicts are less possible. So you can get high performance data transmission with in comparison low baud rates.

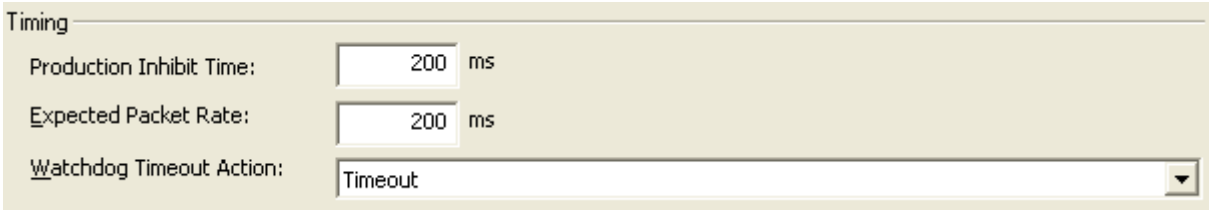


Figure 33: Change of State Connection – Timing

© Hilscher, 2009-2013

The **Production Inhibit Time**, one for each connection, configures the minimum delay time between new data production in multiples of a millisecond. The timer is reloaded each time new data production through the established connection occurs. While the timer is running the device suppresses new data production until the timer has expired. This method prevents that the device is overloaded with to fast incoming requests.

The value 0 defines no **Production Inhibit Time** and data production can and will be done as fast as possible.

The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer. The value is used by the device to reload its 'Transmission Trigger' and 'Watchdog Timer'. In **Change of State** connections the fourfold value of **the Expected Packet Rate** is used to build the 'Watchdog Timer'. If no data production of the remote station takes place within this time, so the connection changes into a watchdog timeout error. Incoming data productions of the remote station load the Watchdog Time again to the fourfold value of the **Expected Packet Rate**.

If a data production did not take place since starting the 'Transmission Trigger Timer' as single values of the **Expected Packet Rate**, so the device triggers a data production at the latest then automatically.

Note: the **Production Inhibit Time** is verified against the **Expected Packet Rate**. If the **Expected Packet Rate** value is unequal to zero, but less than the **Production Inhibit Time** value, then an error message is displayed by the application.

The **Watchdog Timeout Action** defines the device behavior when the watchdog timer in the device (Slave) expires. The following actions are adjustable:

- **Timeout:** The connection transitions to the timeout state and remains in this state until it is Reset or Deleted.
- **Auto reset:** The connection remains in the established state and immediately restarts the Inactivity/Watchdog timer.
- **Auto delete:** The connection class automatically deletes the connection if it experiences an Inactivity/Watchdog timeout.

© Hilscher, 2009-2013

4.4.1.3 Cyclic Connection

If the **Cyclic Connection** type was activated, the elements of this connection type are editable. Otherwise this dialog is disabled.

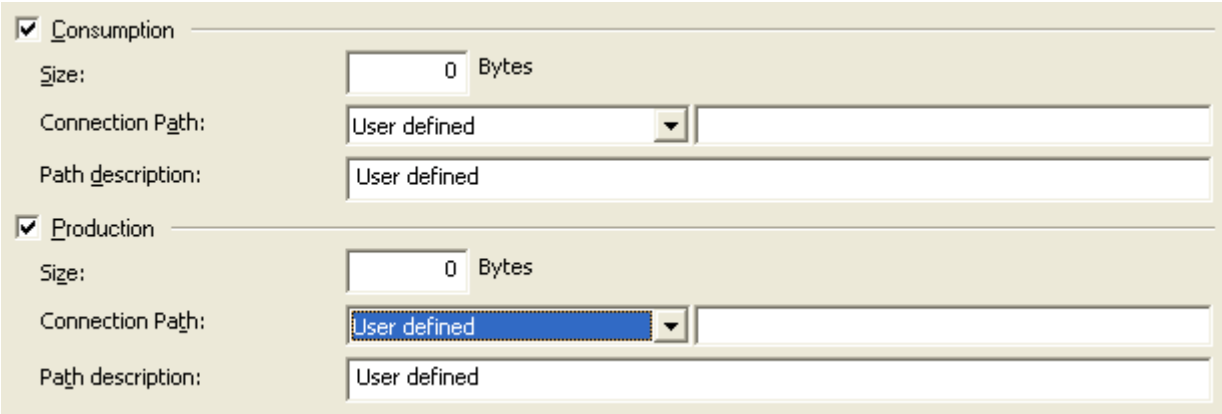


Figure 34: Cyclic Connection - Consumption and Production

At this transmission type a data production takes place automatically, if the 'Transmission Trigger Timer' has expired as single value of the **Expected Packet Rate**.

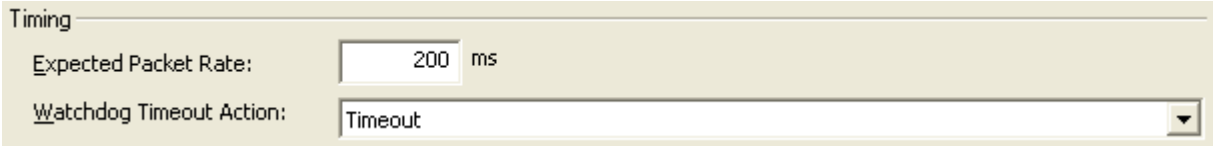


Figure 35: Cyclic Connection - Timing

The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer. The value is used by the device to reload its 'Transmission Trigger' and 'Watchdog Timer'. In **Cyclic** connections the fourfold value of the **Expected Packet Rate** is used to reload the 'Transmission Trigger Timer' and the 'Watchdog Timer'. If no data production of the remote station takes place within this time, so the connection changes into a watchdog timeout error. Incoming data productions of the remote station load the Watchdog Time again to the fourfold value of the **Expected Packet Rate**.

The **Watchdog Timeout Action** defines the device behavior when the watchdog timer in the device (Slave) expires. The following actions are adjustable:

- **Timeout:** The connection transitions to the timeout state and remains in this state until it is Reset or Deleted.
- **Auto reset:** The connection remains in the established state and immediately restarts the Inactivity/Watchdog timer.
- **Auto delete:** The connection class automatically deletes the connection if it experiences an Inactivity/Watchdog timeout.

© Hilscher, 2009-2013

4.4.1.4 Bit-Strobe Connection

If the **Bit-Strobe Connection** type was activated, the elements of this connection type are editable. Otherwise this dialog is disabled.

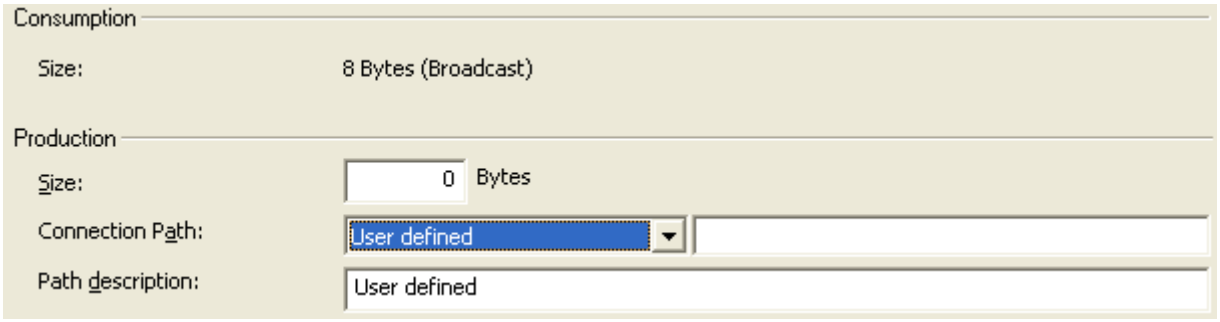


Figure 36: Bit-Strobe Connection - Consumption and Production

Bit strobe command and response messages rapidly move small amounts of output data between the Master device and one/some/all Slave devices. The bit strobe message contains a bit string of 64 bits of output data, one output bit per possible device. Each bit in there is assigned to one device address (MAC-ID) in the network. Herewith this service has broadcast functionality that means more than one Slave device can be addressed by one command. Because all addressed Slave devices get this command at the same time, this command is normally used to synchronize data transfer to several Slave devices. A Slave device can take its corresponding output bit as a real output information to give it to the peripheral connections (e.g. an LED) and/or use the bit as a trigger to send back its input data with a poll response message. The data that can be sent back from each Slave after a bit strobe command was received is limited to 8 bytes in length. Bit strobe usage causes therefore a reduced bus loading than poll connections.

Figure 37: Bit-Strobe Connection - Timing

The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer. The fourfold value is used by the device later to reload its 'Watchdog Timer'. If no data production of the remote station takes place within this time, so the connection changes into a watchdog timeout error. Incoming data productions of the remote station load the Watchdog Time again to the fourfold value of the **Expected Packet Rate**.

© Hilscher, 2009-2013

The **Watchdog Timeout Action** defines the device behavior when the watchdog timer in the device (Slave) expires. The following actions are adjustable:

- **Timeout:** The connection transitions to the timeout state and remains in this state until it is Reset or Deleted.
- **Auto reset:** The connection remains in the established state and immediately restarts the Inactivity/Watchdog timer.
- **Auto delete:** The connection class automatically deletes the connection if it experiences an Inactivity/Watchdog timeout.

To reset the defalut values from the EDS file, click

.

© Hilscher, 2009-2013

4.5 Parameter

In the **Parameter** dialog the parameter data of the device can be edited.

If default parameters are configured in the EDS file for this device, these are inserted automatically. Some of devices need further parameterization data, to change for example a measurement limitation or a value range. These data are device specific and their functionality can not be explained at this point. The explanation can be found in the corresponding device manual.

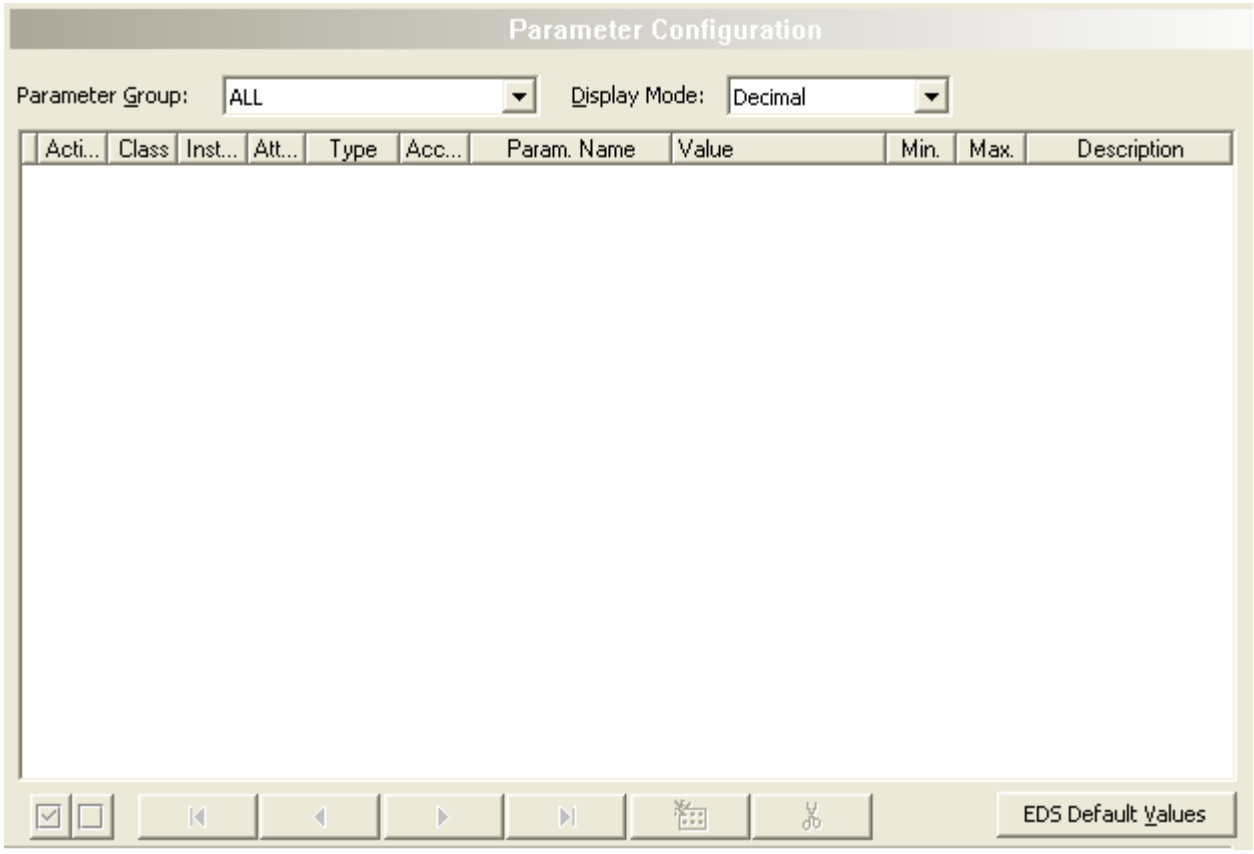


Figure 38: Parameter Configuration

Parameter	Value	Description
Parameter Group	**ALL** USER DEFINED xxx	All parameter groups defined in the EDS file are merged in one table. A parameter group defined by the user is displayed. In the EDS file of the device further parameter groups can be defined, which are also displayed in the pull-down menu Parameter Group. The name of this parameter group itself is also defined in the EDS file.
Display Mode	**Decimal** Hexadecimal	The values **Min** and **Max** in the table are indicated in decimal notation by default. By selecting the **Display Mode** Hexadecimal the values are shown in hexadecimal notation.

Table 20: Change Parameter Data

By using a data set the respective parameter value for the Master configuration are approved and transferred to the Slave by the Master during the initialization phase.

To enable or disable all of the data sets at the same time click .

© Hilscher, 2009-2013

A description of the individual parameters, indicated by **Class**, **Instance** and **Attribute**, can be referred in the device description of the manufacturer.

If “**User Defined**” is selected in **Parameter Group**, the entries in the columns **Param. Name** and **Value** are editable. Otherwise the entries are fixed and can not be changed.

A description on how table lines can be selected, inserted or deleted can be found in section *Table Lines* on pag 16 .

To reset the defalut values from the EDS file, click

.

4.6 Device Settings

At the **Device Settings** pane device related settings can be made. These settings are assigned after a download of the configuration.

Information about the download you find in section *Download Configuration* on page 67).

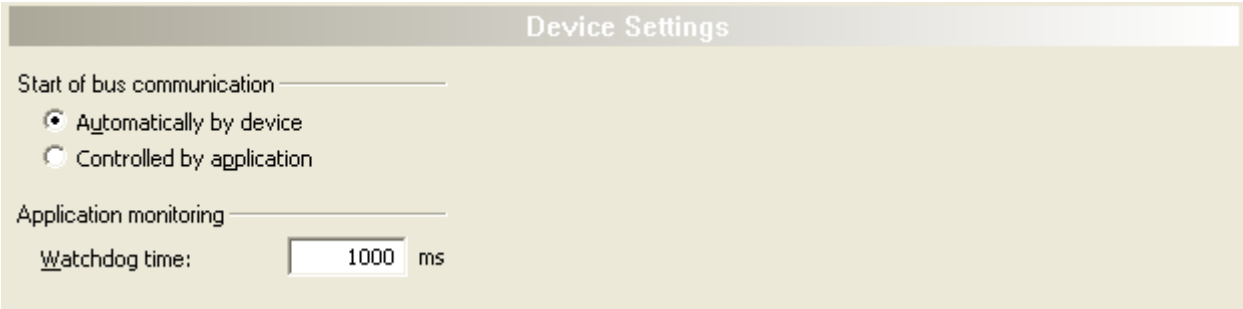


Figure 39: Configuration > Device Settings

Note: The setting options at the dialog pane **Device Settings** for client specific variants of the configuration software can differ from the setting options displayed here.

© Hilscher, 2009-2013

4.6.1 Start of Bus Communication

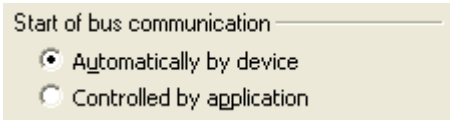


Figure 40: Device Settings > Start of Bus Communication

If **Automatically by device** is selected, the DeviceNet Slave device starts with the data exchange on the bus after the initialization has been ended.

When **Controlled by application** is selected, the application program must activate the data exchange on the bus.

Note: The setting options under **Start of Bus Communication** for client specific variants of the configuration software can differ from the setting options displayed here.

4.6.2 Application Monitoring

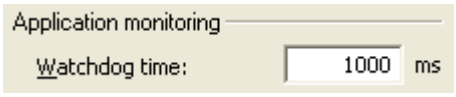


Figure 41: Device Settings > Application Monitoring

The **Watchdog time** determines the time within which the device watchdog must be re-triggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the watchdog is deactivated and the application program monitoring is deactivated too.

The permissible range of values of the watchdog time is 20 to 65535. By default the watchdog time value equals to 1000 ms.

Watchdog time	Range of Value / Value
Permissible range of values	20 … 65535 ms
Default	1000 ms
The software watchdog is deactivated.	0 ms

Table 21: Range of Value / Value for the Watchdog time

Note: The setting options under **Application Monitoring** for client specific variants of the configuration software can differ from the setting options displayed here.

© Hilscher, 2009-2013

5 Device Description

5.1 About Device Description

Dialog Panes "Device Description"

The table below gives an overview for the **Device Description** dialog panes descriptions:

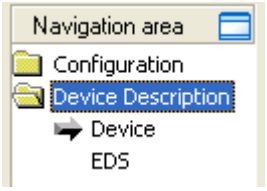
DeviceNet Slave DTM	Folder Name / Section	Page
 *Navigation Area - Description*	*Device*	64
	EDS	65

Table 22: Descriptions of the Dialog Panes Device Description

5.2 Device

The **Device Info** dialog contains manufacturer information about the device, which is defined in the EDS file. The following information is indicated:

Parameter	Meaning
Vendor name	Vendor name of the device
Vendor ID	Identification number of the manufacturer
Product Type	Communication Adapter
Product Type String	Product Name as string
Product Code	Product code of the device
Product Name	Name of the device The variable Product Name is a text string that should represent a short description of the product/product family.
Major Revision	Major Revision
Minor Revision	Minor Revision
Catalog	Used catalog name
Icon file	Used icon file name

Table 23: General Device Information

© Hilscher, 2009-2013

5.3 EDS

The **EDS Viewer** shows the content of the EDS file in a text view.

Under **Filename** the file directory path and the file name of the displayed EDS file is displayed. **Find what** offers a search feature to search for text contents within the text of the EDS file.

In the EDS Viewer window on the left side, the line number is displayed for simple overview, the further entries show the EDS file in text format.

Parameter	Meaning
Filename	File directory path and the file name of the displayed EDS file.
Find what	Search feature to search for text contents within the text of the EDS file.
Match case	Search option
Match whole word	Search option

Table 24: Device Description – EDS Viewer

© Hilscher, 2009-2013

6 Online Functions

6.1 Connecting/Disconnecting Device

Note: Several DeviceNet Slave DTM functions e. g. **Diagnosis** or the configuration download in the FDT Framework require an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device.

Connecting Device

The following steps are needed to establish a connection from the DeviceNet Slave DTM to a DeviceNet Slave device:

Under **Settings** in the **Driver** pane:

1. Select one or several drivers.
2. Configure the driver if necessary.

Under **Settings** in the **Device Assignment** pane:

3. Scan for the devices (with or without firmware).
4. Select the devices (with or without firmware) and apply the selection.

Under **Settings** in the **Firmware Download** pane if not yet a firmware was loaded to the device:

5. Select and download the firmware.

Under **Settings** in the **Device Assignment** pane if not yet a firmware was loaded to the device:

6. Scan for the device (with firmware) once more.
7. Select the device (with firmware) once more.

An overview of the descriptions for these steps you find in the section *Overview Settings* on page 24.

8. In the DTM interface dialog select the **OK** button, to apply the selection and to close the DTM interface dialog.
9. Put a right-click on the DeviceNet Slave device icon.
10. Select the **Connect** command from the context menu.

The DeviceNet Slave device now is connected to the DeviceNet Slave DTM via an online connection. In the network view the device description at the device icon of the Slave device is displayed with a green colored background.

© Hilscher, 2009-2013

Disconnecting Device

To disconnect an online connection from the DeviceNet Slave device to a DeviceNet Slave DTM take the following steps:

1. In the DTM interface dialog select the **OK** button, to close the DTM interface dialog.
2. Right-click on the DeviceNet Slave device icon.
3. Select the **Disconnect** command from the context menu.

In the network view the device description at the device icon of the Master is not any more displayed with a green colored background. Now the DeviceNet Slave device is disconnected from the DTM.

6.2 Download Configuration

Note: To download configuration parameter data to the DeviceNet Slave device an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device is required.

Further information can be found in the section *Connecting/Disconnecting Device* on page 66.

To transfer the configuration with the corresponding parameter data to the DeviceNet Slave device you download the data to it using the frame application of the configuration software.

© Hilscher, 2009-2013

7 Diagnosis

7.1 Overview Diagnosis

The dialog **Diagnosis** serves to diagnose the device behavior and communication errors. For diagnosis the device must reside in online state.

Diagnosis Panes

The table below gives an overview for the individual **Diagnosis** dialog panes descriptions:

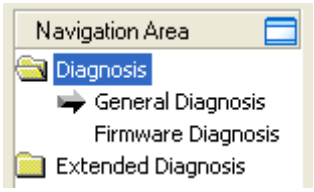
DeviceNet Slave DTM	Folder Name / Section	Manual Page
 *Navigation Area - Diagnosis*	*General Diagnosis*	69
	Firmware Diagnosis	71

Table 25: Descriptions of the Diagnosis Panes

Online Connection to the Device

Note: Accessing the **Diagnosis** panes of the DeviceNet Slave DTM requires an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device. For further information refer to section *Connecting/Disconnecting Device* on page 66.

Important: For a 2-channel device channel 1 or channel 2 each must be connected to the DTM separately.

How to proceed

1. In the Slave DTM diagnosis dialog check whether the communication is OK:

Diagnosis > **General Diagnosis** > **Device status "Communication"** must be green!

2. **"Communication"** is green: Open the **IO Monitor** and test the input or output data..
3. **"Communication"** is not green: Use **Diagnosis** and **Extended diagnosis** for troubleshooting.

Extended Diagnosis

The **Extended Diagnosis** helps to find communication and configuration errors, when default diagnosis fails. For further information refer to section *Overview Extended Diagnosis* on page 72.

© Hilscher, 2009-2013

7.2 General Diagnosis

Information regarding the Device State and other general diagnosis parameters are displayed in the **General Diagnosis** dialog.

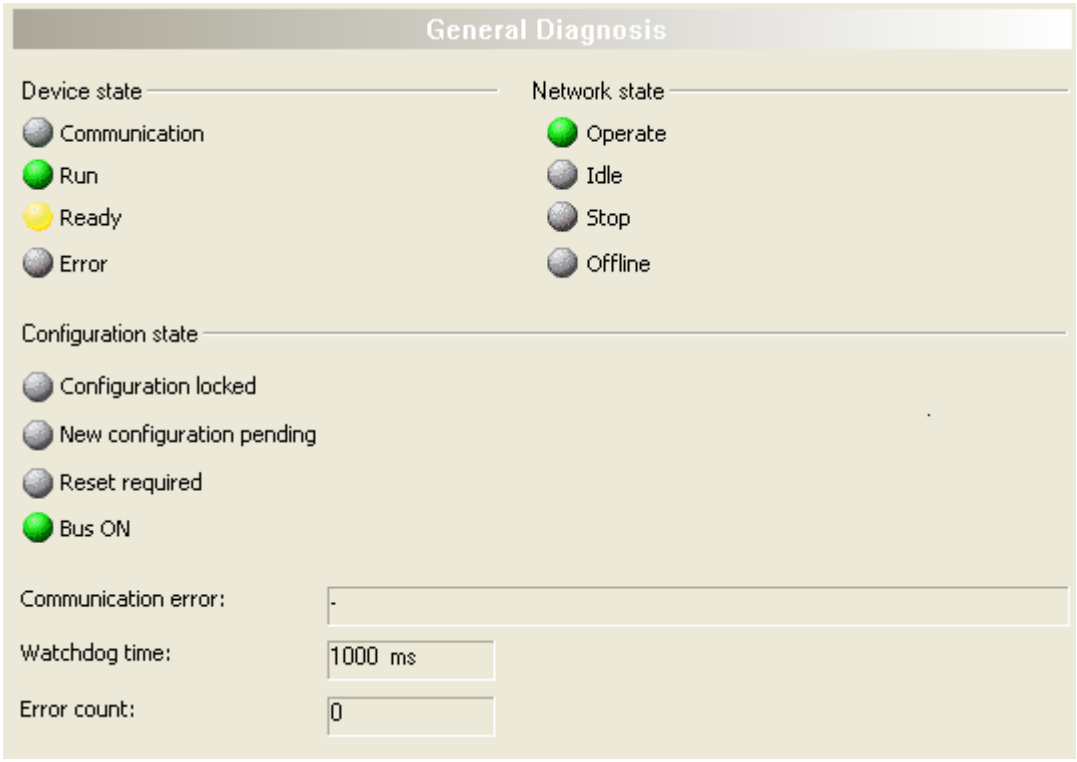


Figure 42: General Diagnosis

LED	Meaning	Color	State
Device State			
Communication	Shows whether the DeviceNet device executes the network communication.	(green)	In COMMUNICATION state
		(gray)	Not in COMMUNICATION state
Run	Shows whether the DeviceNet device has been configured correctly.	(green)	Configuration OK
		(gray)	Configuration not OK
Ready	Shows whether the DeviceNet device has been started correctly. The DeviceNet device waits for a configuration.	(yellow)	Device READY
		(gray)	Device not READY
Error	Shows whether the DeviceNet device records a device status error. For further information about the error characteristics and the number of counted errors refer to the extended diagnosis.	(red)	ERROR
		(gray)	No ERROR
Network State			
Operate	Shows whether the DeviceNet device is in data exchange. In a cyclic data exchange the input data or the output data of the DeviceNet Slave are transmitted to the DeviceNet Master.	(green)	In OPERATION state
		(gray)	Not in OPERATION state
Idle	Shows whether the DeviceNet device is in idle state.	(yellow)	In IDLE state
		(gray)	Not in IDLE state

© Hilscher, 2009-2013

LED	Meaning	Color	State
Stop	Shows whether the DeviceNet device is in Stop state: There is no cyclic data exchange at the DeviceNet network. The DeviceNet device was stopped by the application program or it changed to the Stop state because of a bus error.	(red)	In STOP state
		(gray)	Not in STOP state
Offline	The DeviceNet Slave is offline as long as it does not have a valid configuration.	(yellow)	In OFFLINE state
		(gray)	Not in OFFLINE state
Configuration State			
Configuration locked	Shows whether the DeviceNet device configuration is locked, to avoid the configuration data are typed over.	(yellow)	Configuration LOCKED
		(gray)	Configuration not LOCKED
New Configura-tion pending	Shows whether a new DeviceNet device configuration is available.	(yellow)	New Configuration pending
		(gray)	No new Configuration pending
Reset required	Shows whether a firmware reset is required as a new DeviceNet device configuration has been loaded into the device.	(yellow)	RESET required
		(gray)	No RESET required
Bus ON	Shows whether the bus communication was started or stopped. I. e., whether the device is active on the bus or no bus communication to the device is possible and no response telegrams are sent.	(green)	Bus ON
		(gray)	Bus OFF

Table 26: Indication General Diagnosis

Parameter	Meaning
Communication Error	Shows the name of the communication error. If the cause of error is resolved, the value will be set to zero again.
Watchdog time	Shows the watchdog time in ms.
Error Count	This field holds the total number of errors detected since power-up, respectively after reset. The protocol stack counts all sorts of errors in this field no matter whether they were network related or caused internally.

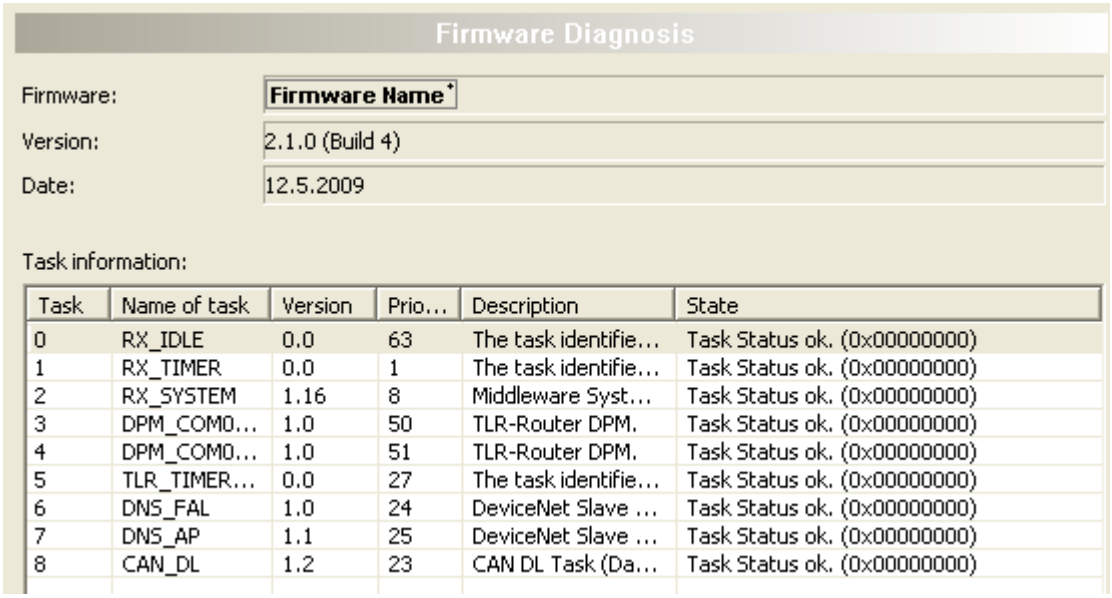
Table 27: Parameter General Diagnosis

© Hilscher, 2009-2013

7.3 Firmware Diagnosis

In the dialog **Firmware Diagnosis** the actual task information of the firmware is displayed.

Under **Firmware** or **Version** the name of the firmware and version (including the date) are indicated.



Task	Name of task	Version	Prio...	Description	State
0	RX_IDLE	0.0	63	The task identifie...	Task Status ok. (0x00000000)
1	RX_TIMER	0.0	1	The task identifie...	Task Status ok. (0x00000000)
2	RX_SYSTEM	1.16	8	Middleware Syst...	Task Status ok. (0x00000000)
3	DPM_COM0...	1.0	50	TLR-Router DPM.	Task Status ok. (0x00000000)
4	DPM_COM0...	1.0	51	TLR-Router DPM.	Task Status ok. (0x00000000)
5	TLR_TIMER...	0.0	27	The task identifie...	Task Status ok. (0x00000000)
6	DNS_FAL	1.0	24	DeviceNet Slave ...	Task Status ok. (0x00000000)
7	DNS_AP	1.1	25	DeviceNet Slave ...	Task Status ok. (0x00000000)
8	CAN_DL	1.2	23	CAN DL Task (Da...	Task Status ok. (0x00000000)

*Figure 43: Firmware Diagnosis (*The name of the Firmware is displayed.)*

Task Information:

The table **Task Information** is listing the task information of the single firmware tasks.

Column	Meaning
Task	Task number
Name of task	Name of the task
Version	Version number of the task
Prio	Priority of the task
Description	Description of the task
Status	Current status of the task

Table 28: Description Table Task Information

© Hilscher, 2009-2013

8 Extended Diagnosis

8.1 Overview Extended Diagnosis

The **Extended Diagnosis** of the DeviceNet Slave DTM helps to find communication and configuration errors. Therefore it contains a list of diagnosis structures as online counter, states and parameters.

Dialog Panes "Extended Diagnosis"

The table below gives an overview for the **Extended Diagnosis** dialog panes descriptions:

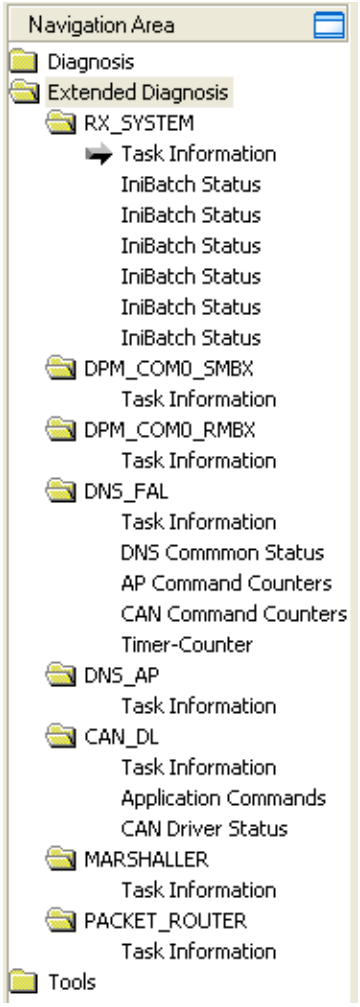
DeviceNet Slave DTM	Folder Name in the Navigation Area	Dialog Pane	Page
 *Navigation Area - Extended Diagnosis*	*RX-SYSTEM*	*Task Information*	*73*
		IniBatch Status	*74*
	DPM_COMO_SMBX	*Task Information*	*73*
	DPM_COMO_RMBX	*Task Information*	*73*
	DNS_FAL	*Task Information*	*73*
		DNS Common Status	*75*
		AP Commands Counter	*76*
		CAN Command Counter	*76*
		Timer Counter	*77*
	DNS_AP	*Task Information*	*73*
	CAN_DL	*Task Information*	*73*
		AP Commands Counter	*78*
		CAN Driver Status	*79*
	MARSHALLER	*Task Information*	*73*
	PACKET_ROUTER	*Task Information*	*73*

Table 29: Descriptions of the Dialog Panes Extended Diagnosis

Online Connection to the Device

Note: Accessing the **Extended Diagnosis** dialog panes of the DeviceNet Slave DTM requires an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device. For further information refer to section *Connecting/Disconnecting Device* on page 66.

© Hilscher, 2009-2013

8.2 Task Information

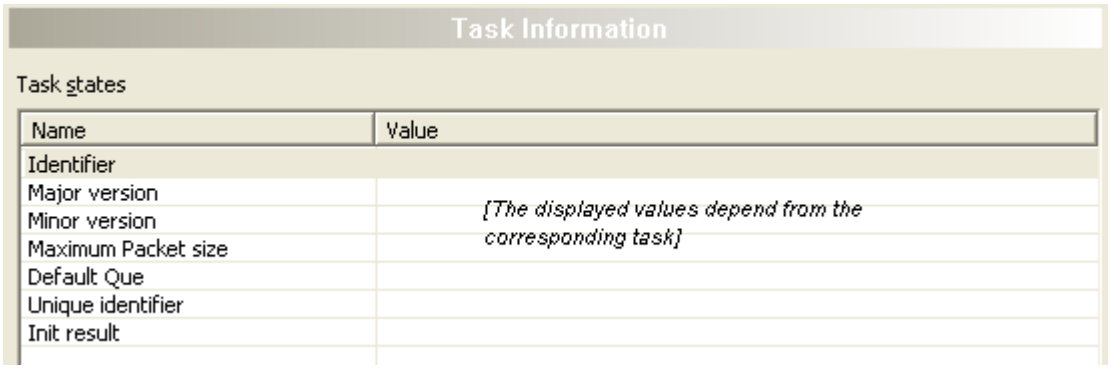


Figure 44: Extended Diagnosis > [Folder Name] > Task Information Example Display

Name	Description
Identifier	Identification number of the task
Major version	Task version, contains incompatible changes
Minor version	Task version, contains compatible changes
Maximum packet size	Maximum packet size, which the task sends
Default Queue	Queue handle, which is accessible via DPM by mailbox.
UUID	Unique user ID, 16 Byte indicator used for task identification and its affiliation e. g. to a stack (therein different identification data are coded in).
Init result	Error Code, 0= no Error The description of the error codes can be found in this manual or in the corresponding software reference manuals.

Table 30: Extended Diagnosis > [Folder Name] > Task Information

© Hilscher, 2009-2013

8.3 IniBatch Status

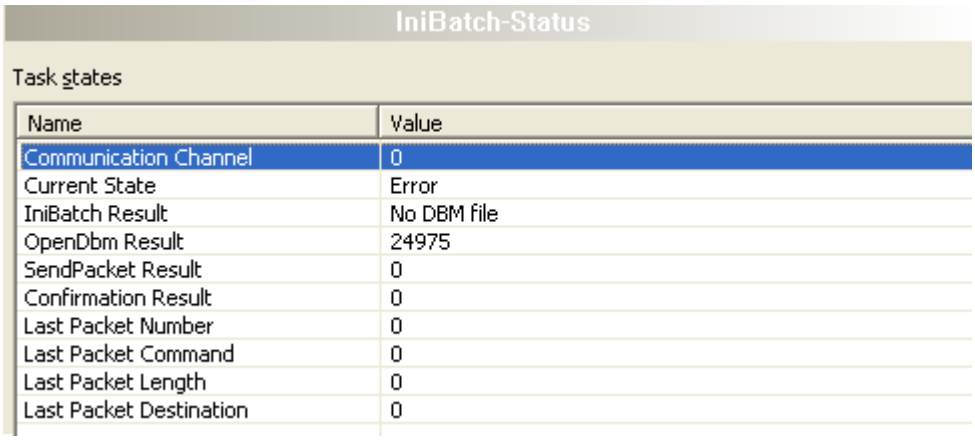


Figure 45: Extended Diagnosis > [Folder Name] > IniBatch Status Example Display

Name	Description
Communication Channel	Number of the communication channel used by the device.
Current State	Idle; IniBatch packets in progress; Retrying to send last packet; Error
IniBatch Result	Ok; No DBM file; No Packet table; No data set available; Data set is shorter than packet length; Packet Buffer is shorter than Packet length; Invalid packet destination; Logical queue not defined Send packet failed; Too many retries; Error in confirmation packet status
OpenDbm Result	Error when opening the IniBatch data base Under "OpenDbm Result" the error code is typed in, when "IniBatch Result" == "No DBM file" (1) is.
SendPacket Result	Error when sending a packet Under "SendPacket Result" the error code is typed in, when "IniBatch Result" == “send packet failed" (8) is.
Confirmation Result	Confirmation error when sending packets Under "Confirmation Result" the packet specific error code from the ulSta is typed in, when "IniBatch Result " == "Error in confirmation packet status" (10) is.
Last Packet Number	Value depends by the communication system.
Last Packet Command	Value depends by the communication system.
Last Packet Length	Value depends by the communication system.
Last Packet Destination	Value depends by the communication system.

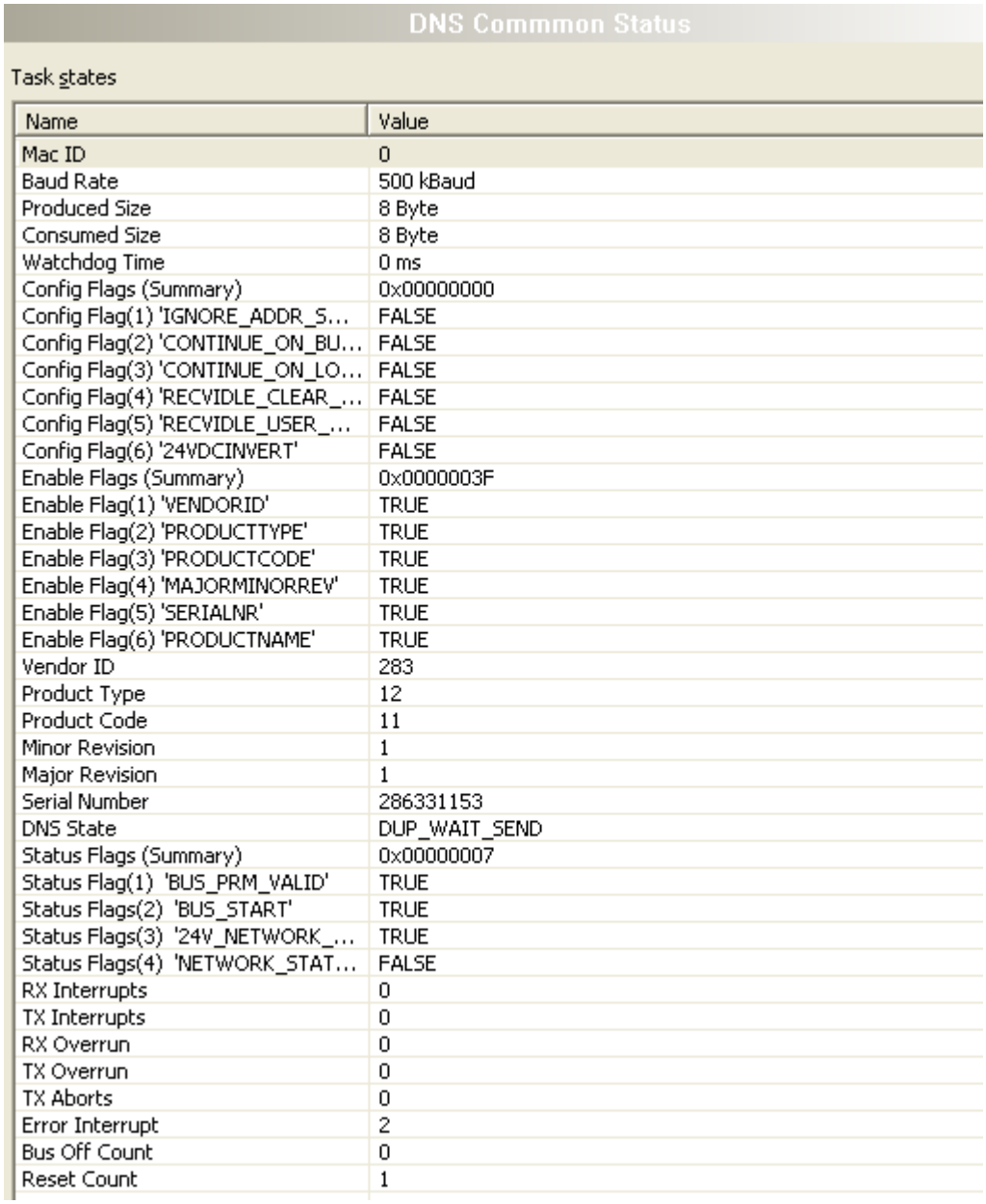
Table 31: Extended Diagnosis > [Folder Name] > IniBatch Status

The task status "Confirmation Result" is bus specific. The other task statuses are rcx-related error codes.

© Hilscher, 2009-2013

8.4 DNS_FAL

8.4.1 DNS Common Status



Name	Value
Mac ID	0
Baud Rate	500 kBaud
Produced Size	8 Byte
Consumed Size	8 Byte
Watchdog Time	0 ms
Config Flags (Summary)	0x00000000
Config Flag(1) 'IGNORE_ADDR_S...	FALSE
Config Flag(2) 'CONTINUE_ON_BU...	FALSE
Config Flag(3) 'CONTINUE_ON_LO...	FALSE
Config Flag(4) 'RECVIDLE_CLEAR_...	FALSE
Config Flag(5) 'RECVIDLE_USER_...	FALSE
Config Flag(6) '24VDCINVERT'	FALSE
Enable Flags (Summary)	0x0000003F
Enable Flag(1) 'VENDORID'	TRUE
Enable Flag(2) 'PRODUCTTYPE'	TRUE
Enable Flag(3) 'PRODUCTCODE'	TRUE
Enable Flag(4) 'MAJORMINORREV'	TRUE
Enable Flag(5) 'SERIALNR'	TRUE
Enable Flag(6) 'PRODUCTNAME'	TRUE
Vendor ID	283
Product Type	12
Product Code	11
Minor Revision	1
Major Revision	1
Serial Number	286331153
DNS State	DUP_WAIT_SEND
Status Flags (Summary)	0x00000007
Status Flag(1) 'BUS_PRM_VALID'	TRUE
Status Flags(2) 'BUS_START'	TRUE
Status Flags(3) '24V_NETWORK_...	TRUE
Status Flags(4) 'NETWORK_STAT...	FALSE
RX Interrupts	0
TX Interrupts	0
RX Overrun	0
TX Overrun	0
TX Aborts	0
Error Interrupt	2
Bus Off Count	0
Reset Count	1

Figure 46: Extended Diagnosis > DNS_FAL > DNS Common Status

Name	Description
[*Status*]	Common DNS Diagnosis status. Indicates the current status of the single tasks. (For further information refer to the API manual.)

Table 32: Extended Diagnosis > DNS_FAL > DNS Common Status

© Hilscher, 2009-2013

8.4.2 AP Commands Counter

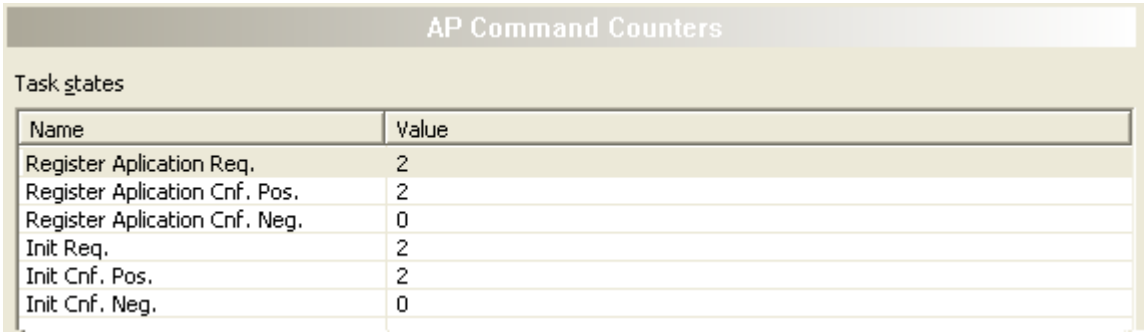


Figure 47: Extended Diagnosis > DNS_FAL > AP Commands Counter

Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the API manual.)

Table 33: Extended Diagnosis > DNS_FAL > AP Commands Counter

8.4.3 CAN Command Counter

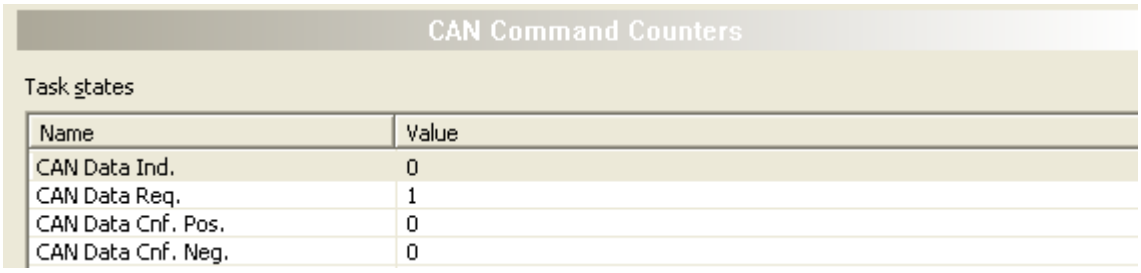


Figure 48: Extended Diagnosis > DNS_FAL > CAN Command Counter

Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the API manual.)

Table 34: Extended Diagnosis > DNS_FAL > CAN Command Counter

© Hilscher, 2009-2013

8.4.4 Timer Counter

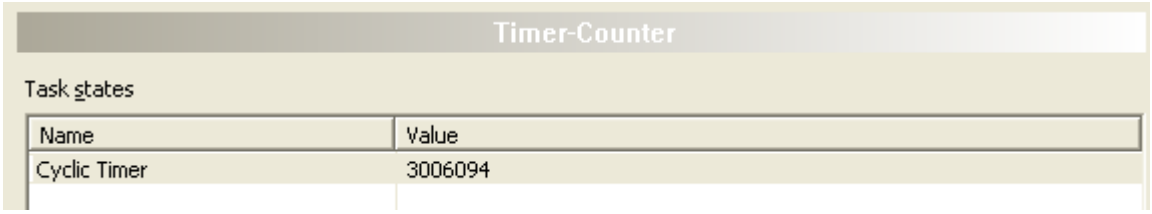


Figure 49: Extended Diagnosis > DNS_FAL > Timer Counter

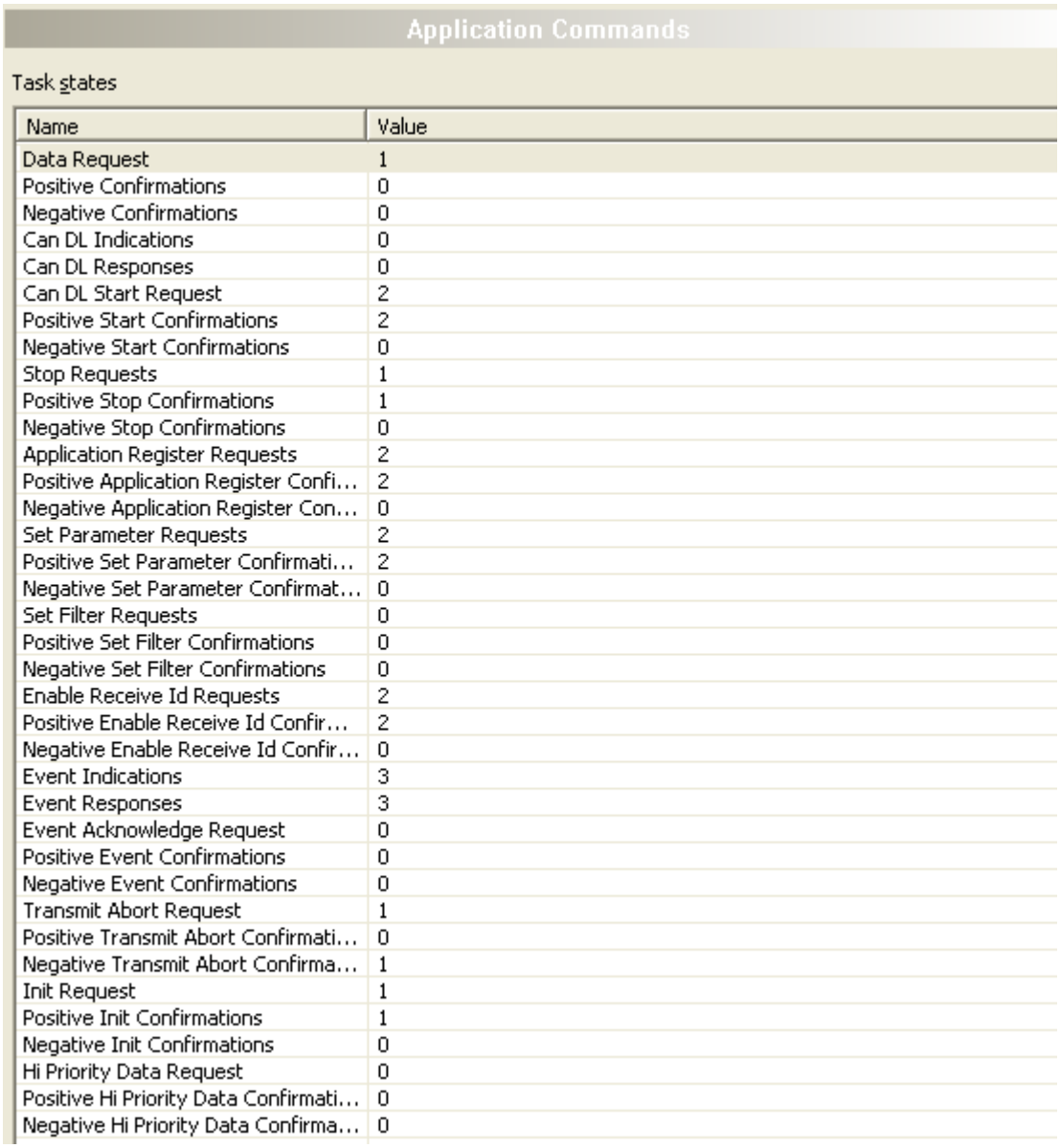
Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the API manual.)

Table 35: Extended Diagnosis > DNS_FAL > Timer Counter

© Hilscher, 2009-2013

8.5 CAN_DL

8.5.1 AP Commands Counter



Name	Value
Data Request	1
Positive Confirmations	0
Negative Confirmations	0
Can DL Indications	0
Can DL Responses	0
Can DL Start Request	2
Positive Start Confirmations	2
Negative Start Confirmations	0
Stop Requests	1
Positive Stop Confirmations	1
Negative Stop Confirmations	0
Application Register Requests	2
Positive Application Register Confi...	2
Negative Application Register Con...	0
Set Parameter Requests	2
Positive Set Parameter Confirmati...	2
Negative Set Parameter Confirmat...	0
Set Filter Requests	0
Positive Set Filter Confirmations	0
Negative Set Filter Confirmations	0
Enable Receive Id Requests	2
Positive Enable Receive Id Confir...	2
Negative Enable Receive Id Confir...	0
Event Indications	3
Event Responses	3
Event Acknowledge Request	0
Positive Event Confirmations	0
Negative Event Confirmations	0
Transmit Abort Request	1
Positive Transmit Abort Confirmati...	0
Negative Transmit Abort Confirma...	1
Init Request	1
Positive Init Confirmations	1
Negative Init Confirmations	0
Hi Priority Data Request	0
Positive Hi Priority Data Confirmati...	0
Negative Hi Priority Data Confirma...	0

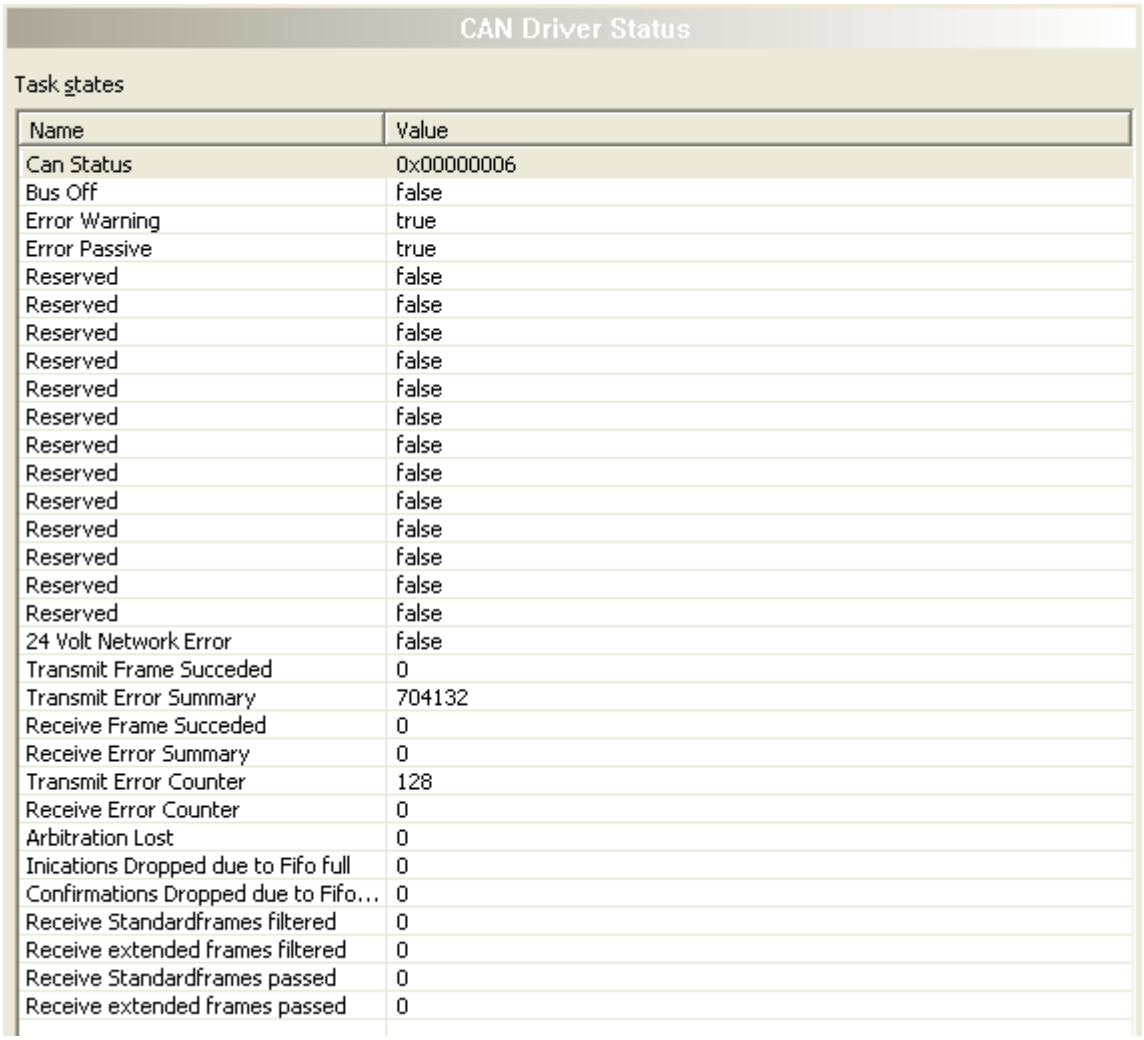
Figure 50: Extended Diagnosis > CAN_DL > AP Commands Counter

Name	Description
[*Service*]	Diagnosis counter of the CAN layer. Indicates the services processed. (The services of the single packets are described in the API manual.)

Table 36: Extended Diagnosis > CAN_DL > AP Commands Counter

© Hilscher, 2009-2013

8.5.2 CAN Driver Status



Name	Value
Can Status	0x00000006
Bus Off	false
Error Warning	true
Error Passive	true
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
24 Volt Network Error	false
Transmit Frame Succeded	0
Transmit Error Summary	704132
Receive Frame Succeded	0
Receive Error Summary	0
Transmit Error Counter	128
Receive Error Counter	0
Arbitration Lost	0
Inications Dropped due to Fifo full	0
Confirmations Dropped due to Fifo...	0
Receive Standardframes filtered	0
Receive extended frames filtered	0
Receive Standardframes passed	0
Receive extended frames passed	0

Figure 51: Extended Diagnosis > CAN_DL > CAN Driver Status

Name	Description
CAN Status Bus Off Error Warning ErrorPassive	Diagnosis status of CAN specific error levels. Indicates the respective status of the CAN bus. (For further information refer to the API manual.)
Reserved	Diagnosis status
24 Volt Network-Error	Diagnosis status of DeviceNet specific error levels. (For further information refer to the API manual.)
[*Service*]	Diagnosis counter of CAN errors. Indicates the services processed. (The services of the single packets are described in the API manual.)

Table 37: Extended Diagnosis > CAN_DL > CAN Driver Status

© Hilscher, 2009-2013

9 Tools

9.1 Overview Tools

Under **Tools** the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes.

Tools Panes

The table below gives an overview for the individual **Tools** dialog panes descriptions:

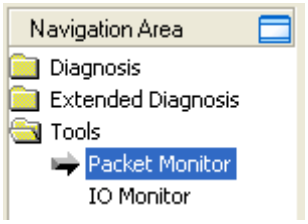
DeviceNet Slave DTM	Folder Name / Section	Manual Page
 *Navigation Area - Tools*	*Packet Monitor*	81
	IO Monitor	89

Table 38: Descriptions of the Diagnosis Panes

Online Connection to the Device

Note: Accessing the **Tools** dialog panes of the DeviceNet Slave DTM requires an online connection from the DeviceNet Slave DTM to the DeviceNet Slave device. For further information refer to section Connecting/Disconnecting Device on page 66.

© Hilscher, 2009-2013

9.2 Packet Monitor

The **Packet Monitor** serves for test and diagnosis purposes.

Data packets, i. e. messages are self-contained blocks of defined data length. The packets are used to communicate with the firmware and they are exchanged between the application (configuration software) and the firmware in the device. Packets can be sent once or cyclically to the connected device controlled by the user and packets received can be displayed.

Data packets comprise from a **Packet Header** and the **Send Data** or from a **Packet Header** and the **Receive Data**. The packet header can be evaluated by the receiver of the packet and contain the sender and receiver address, the data length, an ID number, status and error messages and the command or response code. The minimum packet size amounts 40 Byte for the packet header. The sending and receiving data are added.

For further information to the packet description refer to the *Protocol API Manual*.

➢ Open the **Packet Monitor** via **Tools > Packet Monitor**.

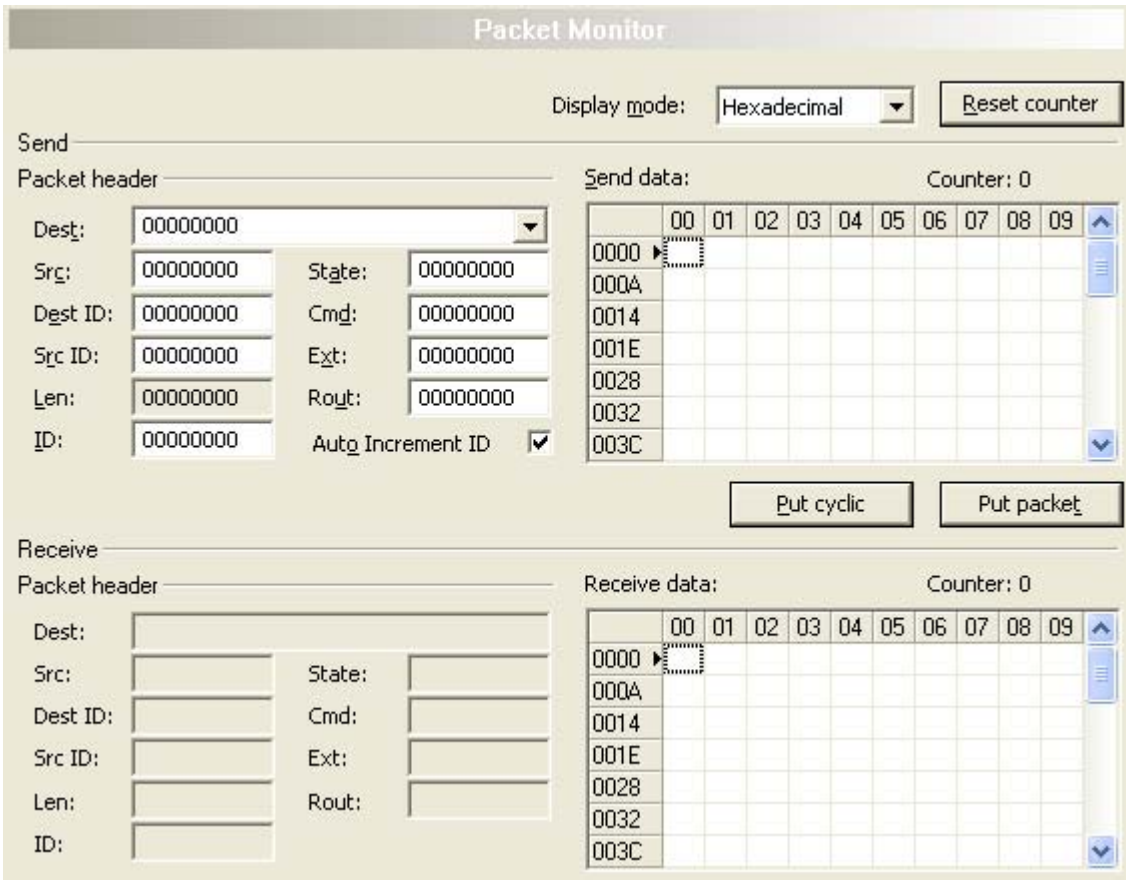


Figure 52: Packet Monitor

Display Mode switches the representation of the send and reception data between decimal and hexadecimal.

➢ Select **Reset Counter** to reset the packet counter.

© Hilscher, 2009-2013

9.2.1 Sending Packet

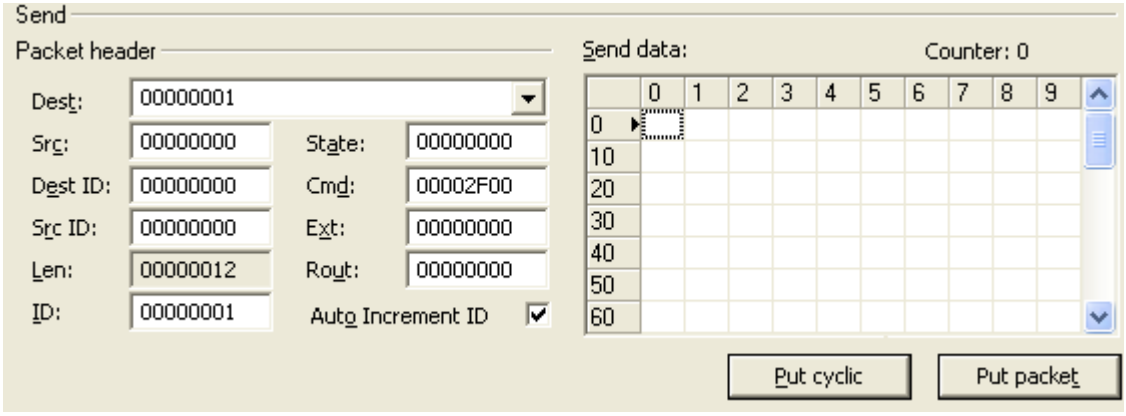


Figure 53: Send > Packet Header and Send Data

Packet Header

Under **Send > Packet Header** the elements of the packet header of the sending packet are displayed, which is transmitted from the application (configuration software) to the device. The packet header of the sending packets contain the elements described in the following table.

Element		Description
Dest	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
Src	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
Dest ID	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
Src ID	Source Queue Reference	Contains an identifier of the sender.
Len	Packet Data Length (in Bytes)	Length of the send respectively receive data.
ID	Packet Identification As Unique Number	Identifies identical data packets among each other.
State	Status / Error Code	Transmits status or error codes to the packet sender.
Cmd	Command / Response Code	Command or respond code.
Ext	Extension	Field for extensions (reserved).
Rout	Routing Information	Internal value of the firmware.

Table 39: Descriptions Packet Header

- Under **Dest** select the receiver (*destination task queue*).
- Under **Cmd** select the command identification *(Request)*.

Auto Increment ID is an increment for the identifier of the data packets and increments the ID by 1 for each newly sent packet.

Send Data

- Under **Send > Send data** enter the send data of the packet, which shall be transmitted from the application (configuration software) to the mailbox of the device. The meaning of the transmitted data depends on the command or response code.

© Hilscher, 2009-2013

Sending Packets once or cyclic

- To send packet once, select **Put packet**.
- To send packet cyclic, select **Put cyclic**.

9.2.2 Receiving Packet

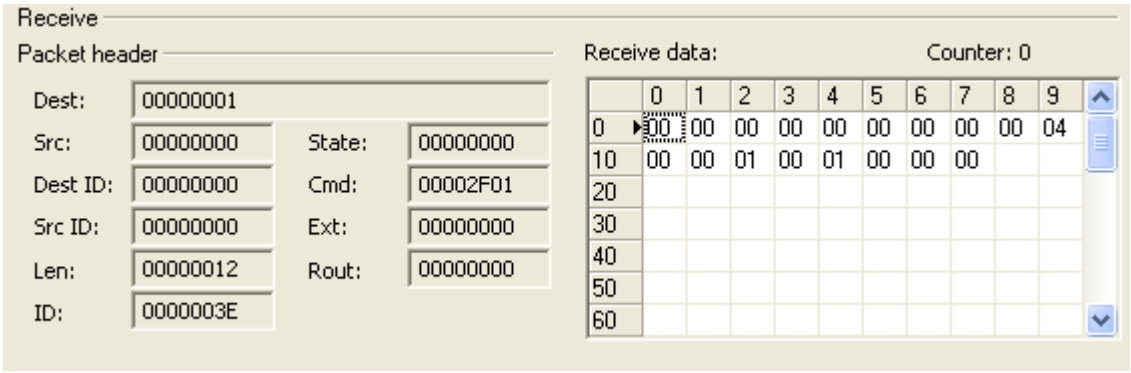


Figure 54: Packet Header and Receive Data

Packet Header

Under **Receive** > **Packet Header** the elements of the packet header of the receiving packet are displayed, which are transmitted back from the device to the application (configuration software). The packet header of the receiving packets contain the elements described in the following table.

Element		Description
Dest	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
Src	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
Dest ID	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
Src ID	Source Queue Reference	Contains an identifier of the sender.
Len	Packet Data Length (in Bytes)	Length of the send respectively receive data.
ID	Packet Identification As Unique Number	Identifies identical data packets among each other.
State	Status / Error Code	Transmits status or error codes to the packet sender.
Cmd	Command / Response Code	Command or respond code.
Ext	Extension	Field for extensions (reserved).
Rout	Routing Information	Internal value of the firmware.

Table 40: Descriptions Packet Header

Receive Data

Under **Receive** > **Receive Data** the receiving data of the packet, which is transmitted back from the device to the application (configuration software) are displayed.

© Hilscher, 2009-2013

9.2.3 Example – „Get DPM I/O Information“

To **read** data from a Slave device via **„Get DPM I/O Information Request“** using the packet monitor, you must proceed as described below. The single steps are explained with help of sample data.

For more information on the steps under **Settings** and **Configuration** refer to the corresponding chapters in this manual. Required information is easy to find via the overview in the section *Configuration Steps* on page 18.

Requirements

The function **„Get DPM I/O Information Request“** can only be used, if:

- The used Slave device supports the **„Get DPM I/O Information Request“** function.
- The used Slave device is operational.

How to proceed

1. In the Slave DTM under **Settings**:

➢ Select the cifX device driver and assign the Slave device.

Alternative to the connection via the cifX device driver via the netX driver an USB, serial or TCP connection can be used.

Packet Description Read Request

Structure Information				
Area	**Variable**	**Type**	**Value / Range**	**Description**
tHead	Structure Information			
A	ulDest	UINT32	0x00000020	Destination Queue Handle CHANNEL
	ulSrc	UINT32	X	Source Queue Handle
	ulDestId	UINT32	0x00000000	Destination Queue Reference
	ulSrcId	UINT32	Y	Source Queue Reference
	ulLen	UINT32	0	Packet Data Length (in Bytes)
	ulId	UINT32	Any	Packet Identification as Unique Number
	ulSta	UINT32	0x00000000	Status
C	ulCmd	UINT32	0x00002F0C	Command Get I/O Data Information
	ulExt	UINT32	0x00000000	Reserved
	ulRout	UINT32	0x00000000	Routing Information

Figure 55: Packet Description GET DPM IO INFO REQUEST

Further information on the packet description for this example are comprised in the *Dual Port Memory Manual,* in section *Get DPM IO Info Request* (e. g. *Revision 09* of the manual, in section *5.3.1*).

© Hilscher, 2009-2013

1. Open Packet Monitor.

➢ **Tools > Packet Monitor**.

2. Enter data for Packet header.

Send > Packet header

➢ Under **Dest** enter or select the receiver.

➢ Under **Cmd** enter the read command identification.

		Sample Data	
A	**Receiver**	*0x20*	(Destination Que Handle)
C	**Read Command Identification**	*00002F0C*	`GET DPM IO INFO` **REQUEST** (Request)

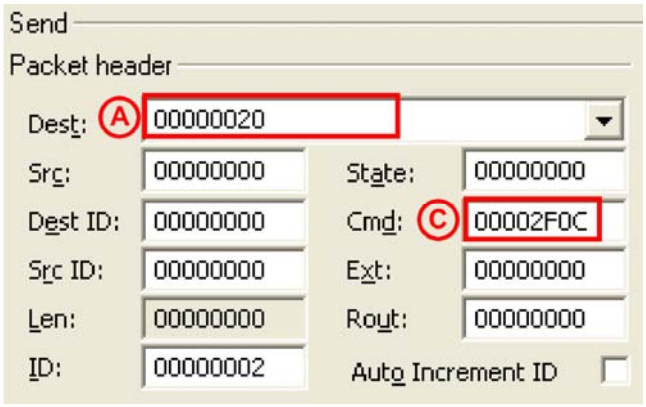


Figure 56: Example - Reading data via „Get DPM IO Info“ – Send > Packet header

Send > Send Data

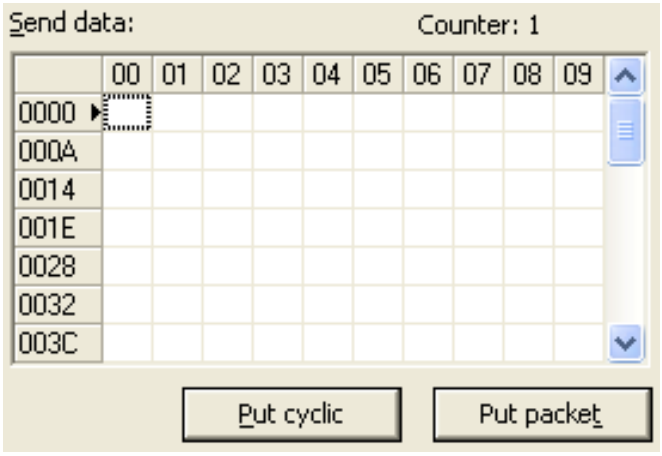


Figure 57: Example - Read data via „Get DPM IO Info“ - Send > Send Data

Note: Entering send data is not required for the **Get DPM I/O Information** packet.

© Hilscher, 2009-2013

3. Sending/Receiving Packets.

➢ To send packet once, select **Put packet**.

➢ The entered values are sent from the packet monitor to the connected Slave device and received packets are displayed.

Packet Description Read Confirmation

Structure Information				
Area	**Variable**	**Type**	**Value / Range**	**Description**
tHead	Structure Information			
A	ulDest	UINT32	From Request	Destination Queue Handle
	ulSrc	UINT32	From Request	Source Queue Handle
	ulDestId	UINT32	From Request	Destination Queue Reference
	ulSrcId	UINT32	From Request	Source Queue Reference
	ulLen	UINT32	4+(20 x n) 0	Packet Data Length (in Bytes) If ulSta = RCX_S_OK Otherwise
	ulId	UINT32	From Request	Packet Identification as Unique Number
	ulSta	UINT32	See Below	Status / Error Code see Section 6
C	ulCmd	UINT32	0x00002F0D	Confirmation Get I/O Data Information
	ulExt	UINT32	0x00000000 0x00000080 0x000000C0 0x00000040	Extension No Sequenced Packet First Packet of Sequence Sequenced Packet Last Packet of Sequence
	ulRout	UINT32	Z	Routing Information, Don't Care, Don't Use
tData	Structure Information			
D	ulNumIOBlock Info	UINT32	0 ... 10	Number n of Block Definitions Below
E	tIoBlock[n]	Array of Structure		I/O Block Definition Structure(s) RCX_DPM_IO_BLOCK_INFO

Packet Structure Reference

```
/* GET DPM I/O INFORMATION CONFIRMATION */
#define RCX_GET_DPM_IO_INFO_CNF              RCX_GET_DPM_IO_INFO_REQ+1

typedef struct RCX_DPM_IO_BLOCK_INFO_Ttag
{
  UINT32   ulSubblockIndex;   /* index of sub block                  */
  UINT32   ulType;            /* type of sub block                   */
  UINT16   usFlags;           /* flags of the sub block              */
  UINT16   usReserved;        /* reserved                            */
  UINT32   ulOffset;          /* offset of I/O data in bytes         */
  UINT32   ulLength;          /* length of I/O data in bytes         */
} RCX_DPM_IO_BLOCK_INFO_T;
```

E

Figure 58: Packet Description GET DPM IO INFO CONFIRMATION

Futher information on the packet description for this example are comprised in the *Dual Port Memory Manual,* in section *Get DPM IO Info Confirmation* (e. g. *Revison 09* of the manual, in section *5.3.2.*) .

© Hilscher, 2009-2013

4. Evaluate received packet.

Receive > Packet header

- Under **Dest** the receiver is displayed.
- Under **State** the status code or possibly an error code is displayed.

All status and error codes are either to be found in this handbook via the section *Overview Error Codes* auf Seite 91, in the *Dual Port Memory Manual* or in the *DeviceNet Slave Protocol API Manual*.

- Under **Cmd** the response identification read request is displayed.

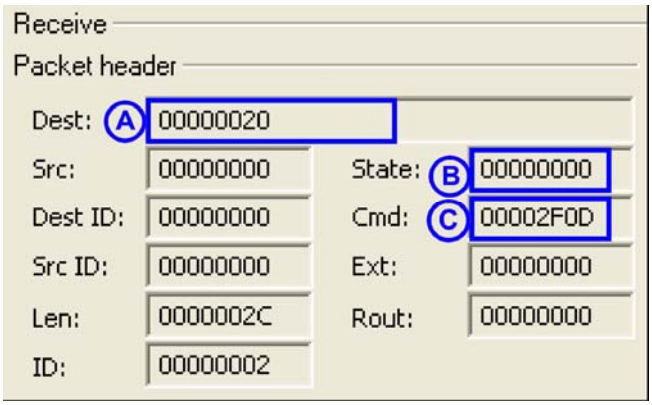


Figure 59: Example - Reading data via „Get DPM IO Info“ – Receive > Packet header

		Sample Data	
A	**Receiver**	*0x20*	(Destination Que Handle)
B	**State**	*00000000*	indicates that the read request could be executed without error. If during the execution of the read request an error was detected, an error code is displayed.
C	**Response Identification Read Request**	*00002F0D*	`GET DPM IO INFO` **`CONFIRMAITON`** (Confirmation)

© Hilscher, 2009-2013

Receive > Receive data

Under **Receive data** the data of the receive packet are displayed.

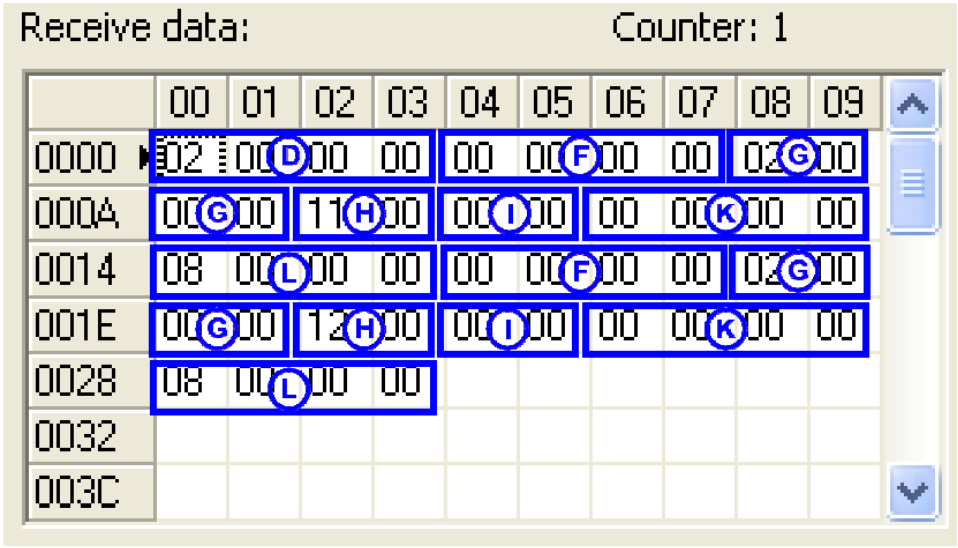


Figure 60: Example - Reading data via „Get DPM IO Info“ – Receive > Receive data

		Range of Value	*Sample Data*	
D	**Num IO Block** (Number n of Block Definitions Below)	0 …10	*02 00 00 00**	
E	**IO Block**[n] (I/O Block Definition Structure(s) RCX_DPM_IO_ BLOCK_INFO)		*Block 0 (IN DPM)*	*Block 1 (OUT DPM)*
F	**Sub block Index** (index of sub block)		*00 00 00 00**	*00 00 00 00**
G	**Type** (type of sub block)	0 … 2^{32} -1	*02 00 00 00**	*02 00 00 00**
H	**Flags** (flags of the sub block)	0 … 65535	*11 00**	*12 00**
I	Reserved	0 … 65535	*00 00**	*00 00**
K	**Offset** (offset of I/O data in bytes)	0 … 2^{32} -1	*00 00 00 00**	*00 00 00 00**
L	**Length** (length of I/O data in bytes)	0 … 2^{32} -1	*08 00 00 00**	*08 00 00 00**
		(UINT32 = 4 Bytes)	**Intel format, e. g. LSB first;*	

The L description shows, that the Slave device is configured with 8 Byte input data and 8 Byte output data.

© Hilscher, 2009-2013

9.3 IO Monitor

The **IO Monitor** serves for test and diagnosis purposes. It provides to view data of the process data image and to change output data easily. The display is always in a Byte manner.

Note: Only change and write output data if you know that no plant disturbances are caused by this. All output data written by the IO Monitor are transmitted at the bus and have effect on subordinate drives, IO etc.

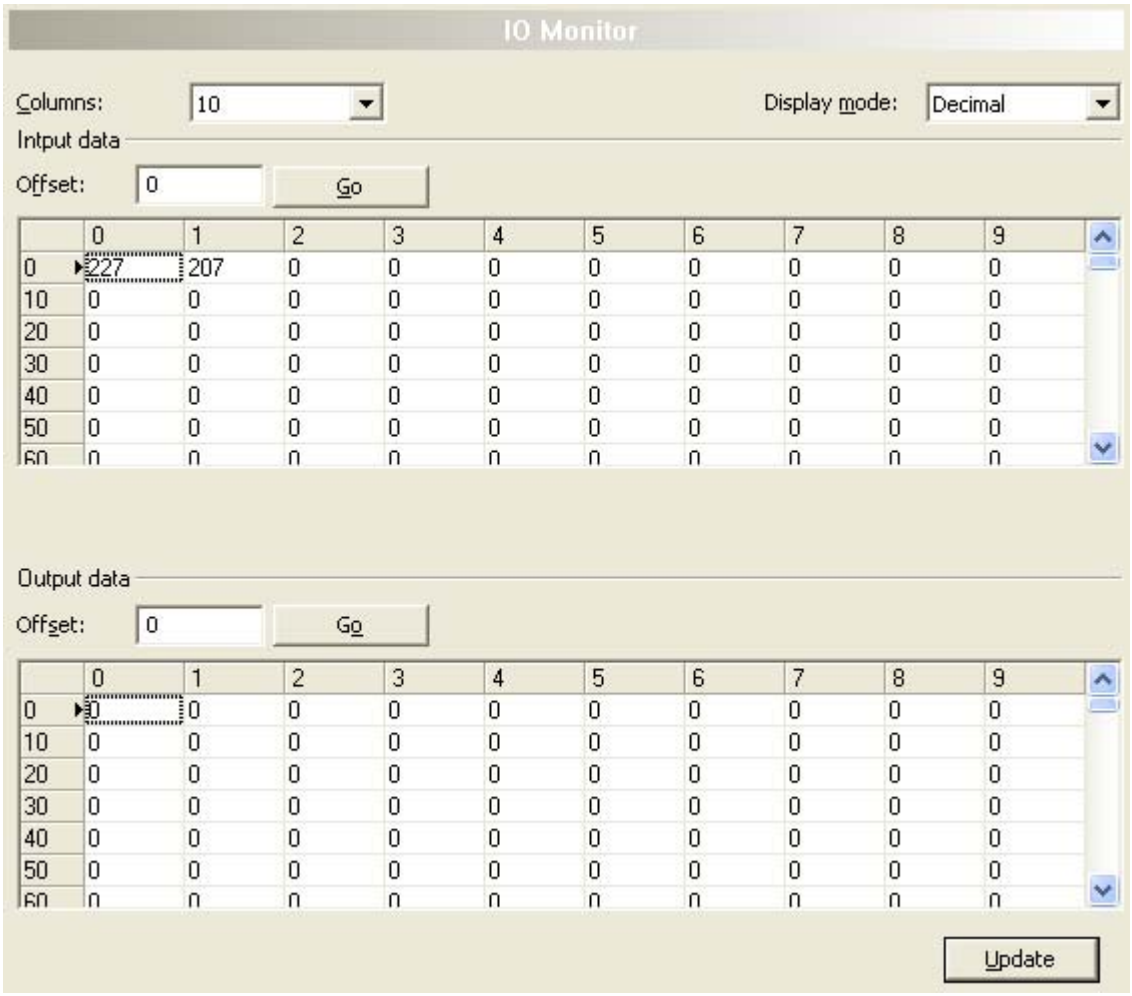


Figure 61: IO Monitor

Columns switches the number of columns.

Display mode switches the representation of the input and output data between decimal and hexadecimal.

Offset / **Go** moves the indication of the data to the entered offset value.

➢ Enter the output value and select **Update**.

↬ Always the data of the process image are displayed, also when these Bytes have not been reserved by the configuration.

© Hilscher, 2009-2013

10 Error Codes

10.1 Error Code Definition

For COM based application, like the ODM Server and ODM drivers, a common error definition is used, similar to the Microsoft Windows® HRESULT definition.

Error Code Structure:

COM Errors are HRESULTs, which are 32 bit values using the following layout:

```
 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+-+-+-----------------------+-------------------------------+
|Sev|C|R|     Facility          |               Code            |
+---+-+-+-----------------------+-------------------------------+


where
 Sev       - is the severity code:
        00 - Success
        01 - Informational
        10 - Warning
        11 - Error
 C         - is the Customer code flag
 R         - is a reserved bit
 Facility  - is the facility code
 Code      - is the facility's status code
```

In this common error definition, several error code regions are already reserved by Windows® itself, the ODM and some other modules.

© Hilscher, 2009-2013

10.2 Overview Error Codes

Overview Error Codes	Range
General Hardware Errors RCX Operating System	*RCX General Task*: 0xC02B0001 to 0xC02B4D52
	RCX Common Status & Errors Codes: 0x00000000 to 0xC002000C
	RCX Status & Error Codes: 0x00000000 to 0xC0000008
ODM Server	*General ODM Error Codes*: 0x8004C700 to 0x8004C761
	General ODM Driver Error Codes: 0x8004C7A0 to 0x8004C7C2
ODM Drivers	*cifX Driver Specific ODM Error*: 0x8004C001 to 0x8004C0A4
cifX Device Driver and netX Driver	*Generic Error*: 0x800A0001 bis 0x800A0017
	Generic Driver Error: 0x800B0001 bis0x800B0042
	Generic Device Error: 0x800C0010 bis0x800C0041
netX Driver	*CIFX API Transport*: 0x800D0001 bis 0x800D0013
	CIFX API Transport Header State Error: 0x800E0001 bis 0x800E000B
DBM	*ODM Error Codes*: 0xC004C810 to 0xC004C878

Table 41: Overview Error Codes and Ranges

The fieldbus specific error codes are described in the manuals of the corresponding protocol tasks.

© Hilscher, 2009-2013

10.3 General Hardware Error Codes

10.3.1 RCX General Task Errors

Error Code (Definition)	Value	Description
RCX_E_QUE_UNKNOWN	0xC02B0001	Unknown Queue
RCX_E_QUE_INDEX_UNKNOWN	0xC02B0002	Unknown Queue Index
RCX_E_TASK_UNKNOWN	0xC02B0003	Unknown Task
RCX_E_TASK_INDEX_UNKNOWN	0xC02B0004	Unknown Task Index
RCX_E_TASK_HANDLE_INVALID	0xC02B0005	Invalid Task Handle
RCX_E_TASK_INFO_IDX_UNKNOWN	0xC02B0006	Unknown Index
RCX_E_FILE_XFR_TYPE_INVALID	0xC02B0007	Invalid Transfer Type
RCX_E_FILE_REQUEST_INCORRECT	0xC02B0008	Invalid File Request
RCX_E_TASK_INVALID	0xC02B000E	Invalid Task
RCX_E_SEC_FAILED	0xC02B001D	Security EEPROM Access Failed
RCX_E_EEPROM_DISABLED	0xC02B001E	EEPROM Disabled
RCX_E_INVALID_EXT	0xC02B001F	Invalid Extension
RCX_E_SIZE_OUT_OF_RANGE	0xC02B0020	Block Size Out Of Range
RCX_E_INVALID_CHANNEL	0xC02B0021	Invalid Channel
RCX_E_INVALID_FILE_LEN	0xC02B0022	Invalid File Length
RCX_E_INVALID_CHAR_FOUND	0xC02B0023	Invalid Character Found
RCX_E_PACKET_OUT_OF_SEQ	0xC02B0024	Packet Out Of Sequence
RCX_E_SEC_NOT_ALLOWED	0xC02B0025	Not Allowed In Current State
RCX_E_SEC_INVALID_ZONE	0xC02B0026	Security EEPROM Invalid Zone
RCX_E_SEC_EEPROM_NOT_AVAIL	0xC02B0028	Security EEPROM Eeprom Not Available
RCX_E_SEC_INVALID_CHECKSUM	0xC02B0029	Security EEPROM Invalid Checksum
RCX_E_SEC_ZONE_NOT_WRITEABLE	0xC02B002A	Security EEPROM Zone Not Writeable
RCX_E_SEC_READ_FAILED	0xC02B002B	Security EEPROM Read Failed
RCX_E_SEC_WRITE_FAILED	0xC02B002C	Security EEPROM Write Failed
RCX_E_SEC_ACCESS_DENIED	0xC02B002D	Security EEPROM Access Denied
RCX_E_SEC_EEPROM_EMULATED	0xC02B002E	Security EEPROM Emulated
RCX_E_INVALID_BLOCK	0xC02B0038	Invalid Block
RCX_E_INVALID_STRUCT_NUMBER	0xC02B0039	Invalid Structure Number
RCX_E_INVALID_CHECKSUM	0xC02B4352	Invalid Checksum
RCX_E_CONFIG_LOCKED	0xC02B4B54	Configuration Locked
RCX_E_SEC_ZONE_NOT_READABLE	0xC02B4D52	Security EEPROM Zone Not Readable

Table 42: RCX General Task Errors

© Hilscher, 2009-2013

10.3.2 RCX Common Status & Errors Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	Success, Status Okay
RCX_E_FAIL	0xC0000001	Fail
RCX_E_UNEXPECTED	0xC0000002	Unexpected
RCX_E_OUTOFMEMORY	0xC0000003	Out Of Memory
RCX_E_UNKNOWN_COMMAND	0xC0000004	Unknown Command
RCX_E_UNKNOWN_DESTINATION	0xC0000005	Unknown Destination
RCX_E_UNKNOWN_DESTINATION_ID	0xC0000006	Unknown Destination ID
RCX_E_INVALID_PACKET_LEN	0xC0000007	Invalid Packet Length
RCX_E_INVALID_EXTENSION	0xC0000008	Invalid Extension
RCX_E_INVALID_PARAMETER	0xC0000009	Invalid Parameter
RCX_E_WATCHDOG_TIMEOUT	0xC000000C	Watchdog Timeout
RCX_E_INVALID_LIST_TYPE	0xC000000D	Invalid List Type
RCX_E_UNKNOWN_HANDLE	0xC000000E	Unknown Handle
RCX_E_PACKET_OUT_OF_SEQ	0xC000000F	Out Of Sequence
RCX_E_PACKET_OUT_OF_MEMORY	0xC0000010	Out Of Memory
RCX_E_QUE_PACKETDONE	0xC0000011	Queue Packet Done
RCX_E_QUE_SENDPACKET	0xC0000012	Queue Send Packet
RCX_E_POOL_PACKET_GET	0xC0000013	Pool Packet Get
RCX_E_POOL_GET_LOAD	0xC0000015	Pool Get Load
RCX_E_REQUEST_RUNNING	0xC000001A	Request Already Running
RCX_E_INIT_FAULT	0xC0000100	Initialization Fault
RCX_E_DATABASE_ACCESS_FAILED	0xC0000101	Database Access Failed
RCX_E_NOT_CONFIGURED	0xC0000119	Not Configured
RCX_E_CONFIGURATION_FAULT	0xC0000120	Configuration Fault
RCX_E_INCONSISTENT_DATA_SET	0xC0000121	Inconsistent Data Set
RCX_E_DATA_SET_MISMATCH	0xC0000122	Data Set Mismatch
RCX_E_INSUFFICIENT_LICENSE	0xC0000123	Insufficient License
RCX_E_PARAMETER_ERROR	0xC0000124	Parameter Error
RCX_E_INVALID_NETWORK_ADDRESS	0xC0000125	Invalid Network Address
RCX_E_NO_SECURITY_MEMORY	0xC0000126	No Security Memory
RCX_E_NETWORK_FAULT	0xC0000140	Network Fault
RCX_E_CONNECTION_CLOSED	0xC0000141	Connection Closed
RCX_E_CONNECTION_TIMEOUT	0xC0000142	Connection Timeout
RCX_E_LONELY_NETWORK	0xC0000143	Lonely Network
RCX_E_DUPLICATE_NODE	0xC0000144	Duplicate Node
RCX_E_CABLE_DISCONNECT	0xC0000145	Cable Disconnected
RCX_E_BUS_OFF	0xC0000180	Network Node Bus Off
RCX_E_CONFIG_LOCKED	0xC0000181	Configuration Locked
RCX_E_APPLICATION_NOT_READY	0xC0000182	Application Not Ready
RCX_E_TIMER_APPL_PACKET_SENT	0xC002000C	Timer App Packet Sent

Table 43: RCX Common Status & Errors Codes

© Hilscher, 2009-2013

10.3.3 RCX Status & Error Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	SUCCESS, STATUS OKAY
RCX_S_QUE_UNKNOWN	0xC02B0001	UNKNOWN QUEUE
RCX_S_QUE_INDEX_UNKNOWN	0xC02B0002	UNKNOWN QUEUE INDEX
RCX_S_TASK_UNKNOWN	0xC02B0003	UNKNOWN TASK
RCX_S_TASK_INDEX_UNKNOWN	0xC02B0004	UNKNOWN TASK INDEX
RCX_S_TASK_HANDLE_INVALID	0xC02B0005	INVALID TASK HANDLE
RCX_S_TASK_INFO_IDX_UNKNOWN	0xC02B0006	UNKNOWN INDEX
RCX_S_FILE_XFR_TYPE_INVALID	0xC02B0007	INVALID TRANSFER TYPE
RCX_S_FILE_REQUEST_INCORRECT	0xC02B0008	INVALID FILE REQUEST
RCX_S_UNKNOWN_DESTINATION	0xC0000005	UNKNOWN DESTINATION
RCX_S_UNKNOWN_DESTINATION_ID	0xC0000006	UNKNOWN DESTINATION ID
RCX_S_INVALID_LENGTH	0xC0000007	INVALID LENGTH
RCX_S_UNKNOWN_COMMAND	0xC0000004	UNKNOWN COMMAND
RCX_S_INVALID_EXTENSION	0xC0000008	INVALID EXTENSION

Table 44: RCX Status & Error Codes

10.3.3.1 RCX Status & Error Codes Slave State

Error Code (Definition)	Value	Description
RCX_SLAVE_STATE_UNDEFINED	0x00000000	UNDEFINED
RCX_SLAVE_STATE_OK	0x00000001	OK
RCX_SLAVE_STATE_FAILED	0x00000002	FAILED (at least one slave)

Table 45: RCX Status & Error Codes Slave State

© Hilscher, 2009-2013

10.4 ODM Error Codes

10.4.1 General ODM Error Codes

Error Code (Definition)	Value	Description
CODM3_E_INTERNALERROR	0x8004C700	Internal ODM Error
ODM3_E_DESCRIPTION_NOTFOUND	0x8004C701	Description not found in ODM database
CODM3_E_WRITEREGISTRY	0x8004C710	Error writing to the registry
CODM3_E_BAD_REGULAR_EXPRESSION	0x8004C711	Invalid regular expression
CODM3_E_COMCATEGORIE_MANAGER_ FAILED	0x8004C712	Component Category Manager could not be instantiated
CODM3_E_COMCATEGORIE_ENUMERATION_ FAILED	0x8004C713	Driver could not be enumerated by the Category Manager
CODM3_E_CREATE_LOCAL_BUFFER	0x8004C714	Error creating local buffers
CODM3_E_UNKNOWNHANDLE	0x8004C715	Unknown handle
CODM3_E_QUEUE_LIMIT_REACHED	0x8004C717	Queue size limit for connection reached
CODM3_E_DATASIZE_ZERO	0x8004C718	Zero data length passed
CODM3_E_INVALID_DATA	0x8004C719	Invalid data content
CODM3_E_INVALID_MODE	0x8004C71A	Invalid mode
CODM3_E_DATABASE_READ	0x8004C71B	Error reading database
CODM3_E_CREATE_DEVICE_THREAD	0x8004C750	Error creating device thread
CODM3_E_CREATE_DEVICE_THREAD_STOP_ EVENT	0x8004C751	Error creating device thread stop event
CODM3_E_CLIENT_NOT_REGISTERED	0x8004C752	Client is not registered at the ODM
CODM3_E_NO_MORE_CLIENTS	0x8004C753	Maximum number of clients reached
CODM3_E_MAX_CLIENT_CONNECTIONS_ REACHED	0x8004C754	Maximum number of client connections reached
CODM3_E_ENTRY_NOT_FOUND	0x8004C755	Driver/device not found
CODM3_E_DRIVER_NOT_FOUND	0x8004C757	The requested driver is unknown to the ODM
CODM3_E_DEVICE_ALREADY_LOCKED	0x8004C758	Device is locked by another process
CODM3_E_DEVICE_UNLOCKED_FAILED	0x8004C759	Device could not be unlocked, lock was set by another process
CODM3_E_DEVICE_LOCK_NECCESSARY	0x8004C75A	Operation requires a device lock to be set
CODM3_E_DEVICE_SUBSCRIPTIONLIMIT	0x8004C75B	Maximum number of servers registered for this device reached
CODM3_E_DEVICE_NOTSUBSCRIBED	0x8004C75C	Process is not registered as a server on this device
CODM3_E_DEVICE_NO_MESSAGE	0x8004C75D	No message available
CODM3_E_TRANSFERTIMEOUT	0x8004C760	Message transfer timeout
CODM3_E_MESSAGE_INSERVICE	0x8004C761	Message in service

Table 46: ODM Error Codes - General ODM Error Codes

© Hilscher, 2009-2013

10.4.2 General ODM Driver Error Codes

Error Code (Definition)	Value	Description
CODM3_E_DRV_OPEN_DEVICE	0x8004C7A0	Packet type unsupported by driver
CODM3_E_DRV_INVALID_IDENTIFIER	0x8004C7A1	Invalid device identifier
CODM3_E_DRV_DEVICE_PARAMETERS_ MISMATCH	0x8004C7A3	Parameters differ from requested device
CODM3_E_DRV_BROWSE_NO_DEVICES	0x8004C7A4	No devices found
CODM3_E_DRV_CREATE_DEVICE_INST	0x8004C7A5	Device instance could not be created
CODM3_E_DRV_DEVICE_NOMORE_TX	0x8004C7A6	Device connection limit reached
CODM3_E_DRV_DEVICE_DUPLICATE_TX	0x8004C7A7	Duplicate transmitter ID
CODM3_E_DRV_DEVICE_NOT_CONFIGURED	0x8004C7A8	Device is not configured
CODM3_E_DRV_DEVICE_COMMUNICATION	0x8004C7A9	Device communication error
CODM3_E_DRV_DEVICE_NO_MESSAGE	0x8004C7AA	No message available
CODM3_E_DRV_DEVICE_NOT_READY	0x8004C7AB	Device not ready
CODM3_E_DRV_INVALIDCONFIGURATION	0x8004C7AC	Invalid driver configuration
CODM3_E_DRV_DLINVALIDMODE	0x8004C7C0	Invalid download mode
CODM3_E_DRV_DLINPROGRESS	0x8004C7C1	Download is active
CODM3_E_DRV_ULINPROGRESS	0x8004C7C2	Upload is active

Table 47: ODM Error Codes - General ODM Driver Error Codess

© Hilscher, 2009-2013

10.4.3 cifX Driver Specific ODM Error Codes

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_BOARD_NOT_INITIALIZED	0x8004C001	DRIVER Board not initialized
DRV_E_INIT_STATE_ERROR	0x8004C002	DRIVER Error in internal init state
DRV_E_READ_STATE_ERROR	0x8004C003	DRIVER Error in internal read state
DRV_E_CMD_ACTIVE	0x8004C004	DRIVER Command on this channel is active
DRV_E_PARAMETER_UNKNOWN	0x8004C005	DRIVER Unknown parameter in function
DRV_E_WRONG_DRIVER_VERSION	0x8004C006	DRIVER Version is incompatible with DLL
DRV_E_PCI_SET_CONFIG_MODE	0x8004C007	DRIVER Error during PCI set configuration mode
DRV_E_PCI_READ_DPM_LENGTH	0x8004C008	DRIVER Could not read PCI dual port memory length
DRV_E_PCI_SET_RUN_MODE	0x8004C009	DRIVER Error during PCI set run mode
DRV_E_DEV_DPM_ACCESS_ERROR	0x8004C00A	DEVICE Dual port ram not accessable(board not found)
DRV_E_DEV_NOT_READY	0x8004C00B	DEVICE Not ready (ready flag failed)
DRV_E_DEV_NOT_RUNNING	0x8004C00C	DEVICE Not running (running flag failed)
DRV_E_DEV_WATCHDOG_FAILED	0x8004C00D	DEVICE Watchdog test failed
DRV_E_DEV_OS_VERSION_ERROR	0x8004C00E	DEVICE Signals wrong OS version
DRV_E_DEV_SYSERR	0x8004C00F	DEVICE Error in dual port flags
DRV_E_DEV_MAILBOX_FULL	0x8004C010	DEVICE Send mailbox is full
DRV_E_DEV_PUT_TIMEOUT	0x8004C011	DEVICE PutMessage timeout
DRV_E_DEV_GET_TIMEOUT	0x8004C012	DEVICE GetMessage timeout
DRV_E_DEV_GET_NO_MESSAGE	0x8004C013	DEVICE No message available
DRV_E_DEV_RESET_TIMEOUT	0x8004C014	DEVICE RESET command timeout
DRV_E_DEV_NO_COM_FLAG	0x8004C015	DEVICE COM-flag not set. Check if Bus is running
DRV_E_DEV_EXCHANGE_FAILED	0x8004C016	DEVICE I/O data exchange failed
DRV_E_DEV_EXCHANGE_TIMEOUT	0x8004C017	DEVICE I/O data exchange timeout
DRV_E_DEV_COM_MODE_UNKNOWN	0x8004C018	DEVICE I/O data mode unknown
DRV_E_DEV_FUNCTION_FAILED	0x8004C019	DEVICE Function call failed
DRV_E_DEV_DPMSIZE_MISMATCH	0x8004C01A	DEVICE DPM size differs from configuration
DRV_E_DEV_STATE_MODE_UNKNOWN	0x8004C01B	DEVICE State mode unknown
DRV_E_DEV_HW_PORT_IS_USED	0x8004C01C	DEVICE Output port already in use
DRV_E_USR_OPEN_ERROR	0x8004C01E	USER Driver not opened (device driver not loaded)
DRV_E_USR_INIT_DRV_ERROR	0x8004C01F	USER Can't connect to device
DRV_E_USR_NOT_INITIALIZED	0x8004C020	USER Board not initialized (DevInitBoard not called)
DRV_E_USR_COMM_ERR	0x8004C021	USER IOCTRL function failed
DRV_E_USR_DEV_NUMBER_INVALID	0x8004C022	USER Parameter DeviceNumber invalid
DRV_E_USR_INFO_AREA_INVALID	0x8004C023	USER Parameter InfoArea unknown
DRV_E_USR_NUMBER_INVALID	0x8004C024	USER Parameter Number invalid
DRV_E_USR_MODE_INVALID	0x8004C025	USER Parameter Mode invalid
DRV_E_USR_MSG_BUF_NULL_PTR	0x8004C026	USER NULL pointer assignment
DRV_E_USR_MSG_BUF_TOO_SHORT	0x8004C027	USER Message buffer too small

© Hilscher, 2009-2013

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_USR_SIZE_INVALID	0x8004C028	USER Parameter Size invalid
DRV_E_USR_SIZE_ZERO	0x8004C02A	USER Parameter Size with zero length
DRV_E_USR_SIZE_TOO_LONG	0x8004C02B	USER Parameter Size too long
DRV_E_USR_DEV_PTR_NULL	0x8004C02C	USER Device address null pointer
DRV_E_USR_BUF_PTR_NULL	0x8004C02D	USER Pointer to buffer is a null pointer
DRV_E_USR_SENDSIZE_TOO_LONG	0x8004C02E	USER Parameter SendSize too large
DRV_E_USR_RECVSIZE_TOO_LONG	0x8004C02F	USER Parameter ReceiveSize too large
DRV_E_USR_SENDBUF_PTR_NULL	0x8004C030	USER Pointer to send buffer is a null pointer
DRV_E_USR_RECVBUF_PTR_NULL	0x8004C031	USER Pointer to receive buffer is a null pointer
DRV_E_DMA_INSUFF_MEM	0x8004C032	DMA Memory allocation error
DRV_E_DMA_TIMEOUT_CH4	0x8004C033	DMA Read I/O timeout
DRV_E_DMA_TIMEOUT_CH5	0x8004C034	DMA Write I/O timeout
DRV_E_DMA_TIMEOUT_CH6	0x8004C035	DMA PCI transfer timeout
DRV_E_DMA_TIMEOUT_CH7	0x8004C036	DMA Download timeout
DRV_E_DMA_DB_DOWN_FAIL	0x8004C037	DMA Database download failed
DRV_E_DMA_FW_DOWN_FAIL	0x8004C038	DMA Firmware download failed
DRV_E_CLEAR_DB_FAIL	0x8004C039	DMA Clear database on the device failed
DRV_E_DEV_NO_VIRTUAL_MEM	0x8004C03C	DMA USER Virtual memory not available
DRV_E_DEV_UNMAP_VIRTUAL_MEM	0x8004C03D	DMA USER Unmap virtual memory failed
DRV_E_GENERAL_ERROR	0x8004C046	DRIVER General error
DRV_E_DMA_ERROR	0x8004C047	DRIVER General DMA error
DRV_E_WDG_IO_ERROR	0x8004C048	DRIVER I/O WatchDog failed
DRV_E_WDG_DEV_ERROR	0x8004C049	DRIVER Device Watchdog failed
DRV_E_USR_DRIVER_UNKNOWN	0x8004C050	USER Driver unknown
DRV_E_USR_DEVICE_NAME_INVALID	0x8004C051	USER Device name invalid
DRV_E_USR_DEVICE_NAME_UKNOWN	0x8004C052	USER Device name unknown
DRV_E_USR_DEVICE_FUNC_NOTIMPL	0x8004C053	USER Device function not implemented
DRV_E_USR_FILE_OPEN_FAILED	0x8004C064	USER File could not be opened
DRV_E_USR_FILE_SIZE_ZERO	0x8004C065	USER File size zero
DRV_E_USR_FILE_NO_MEMORY	0x8004C066	USER Not enough memory to load file
DRV_E_USR_FILE_READ_FAILED	0x8004C067	USER File read failed
DRV_E_USR_INVALID_FILETYPE	0x8004C068	USER File type invalid
DRV_E_USR_FILENAME_INVALID	0x8004C069	USER Invalid filename
DRV_E_FW_FILE_OPEN_FAILED	0x8004C06E	USER Firmware file could not be opened
DRV_E_FW_FILE_SIZE_ZERO	0x8004C06F	USER Not enough memory to load firmware file
DRV_E_FW_FILE_NO_MEMORY	0x8004C070	USER Not enough memory to load firmware file
DRV_E_FW_FILE_READ_FAILED	0x8004C071	USER Firmware file read failed
DRV_E_FW_INVALID_FILETYPE	0x8004C072	USER Firmware file type invalid
DRV_E_FW_FILENAME_INVALID	0x8004C073	USER Firmware file name not valid
DRV_E_FW_DOWNLOAD_ERROR	0x8004C074	USER Firmware file download error
DRV_E_FW_FILENAME_NOT_FOUND	0x8004C075	USER Firmware file not found in the internal table
DRV_E_FW_BOOTLOADER_ACTIVE	0x8004C076	USER Firmware file BOOTLOADER active

© Hilscher, 2009-2013

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_FW_NO_FILE_PATH	0x8004C077	USER Firmware file no file path
DRV_E_CF_FILE_OPEN_FAILED	0x8004C078	USER Configuration file could not be opend
DRV_E_CF_FILE_SIZE_ZERO	0x8004C079	USER Configuration file size zero
DRV_E_CF_FILE_NO_MEMORY	0x8004C07A	USER Not enough memory to load configuration file
DRV_E_CF_FILE_READ_FAILED	0x8004C07B	USER Configuration file read failed
DRV_E_CF_INVALID_FILETYPE	0x8004C07C	USER Configuration file type invalid
DRV_E_CF_FILENAME_INVALID	0x8004C07D	USER Configuration file name not valid
DRV_E_CF_DOWNLOAD_ERROR	0x8004C07E	USER Configuration file download error
DRV_E_CF_FILE_NO_SEGMENT	0x8004C07F	USER No flash segment in the configuration file
DRV_E_CF_DIFFERS_FROM_DBM	0x8004C080	USER Configuration file differs from database
DRV_E_DBM_SIZE_ZERO	0x8004C083	USER Database size zero
DRV_E_DBM_NO_MEMORY	0x8004C084	USER Not enough memory to upload database
DRV_E_DBM_READ_FAILED	0x8004C085	USER Database read failed
DRV_E_DBM_NO_FLASH_SEGMENT	0x8004C086	USER Database segment unknown
DEV_E_CF_INVALID_DESCRIPT_VERSION	0x8004C096	CONFIG Version of the descript table invalid
DEV_E_CF_INVALID_INPUT_OFFSET	0x8004C097	CONFIG Input offset is invalid
DEV_E_CF_NO_INPUT_SIZE	0x8004C098	CONFIG Input size is 0
DEV_E_CF_MISMATCH_INPUT_SIZE	0x8004C099	CONFIG Input size does not match configuration
DEV_E_CF_INVALID_OUTPUT_OFFSET	0x8004C09A	CONFIG Invalid output offset
DEV_E_CF_NO_OUTPUT_SIZE	0x8004C09B	CONFIG Output size is 0
DEV_E_CF_MISMATCH_OUTPUT_SIZE	0x8004C09C	CONFIG Output size does not match configuration
DEV_E_CF_STN_NOT_CONFIGURED	0x8004C09D	CONFIG Station not configured
DEV_E_CF_CANNOT_GET_STN_CONFIG	0x8004C09E	CONFIG Cannot get the Station configuration
DEV_E_CF_MODULE_DEF_MISSING	0x8004C09F	CONFIG Module definition is missing
DEV_E_CF_MISMATCH_EMPTY_SLOT	0x8004C0A0	CONFIG Empty slot mismatch
DEV_E_CF_MISMATCH_INPUT_OFFSET	0x8004C0A1	CONFIG Input offset mismatch
DEV_E_CF_MISMATCH_OUTPUT_OFFSET	0x8004C0A2	CONFIG Output offset mismatch
DEV_E_CF_MISMATCH_DATA_TYPE	0x8004C0A3	CONFIG Data type mismatch
DEV_E_CF_MODULE_DEF_MISSING_NO_SI	0x8004C0A4	CONFIG Module definition is missing,(no Slot/Idx)

Table 48: cifX Driver Specific ODM Error Codes

© Hilscher, 2009-2013

10.5 Error Codes cifX Device Driver and netX Driver

10.5.1 Generic Error Codes

Error Code (Definition)	Value	Description
CIFX_INVALID_POINTER	0x800A0001L	Invalid pointer (NULL) passed to driver
CIFX_INVALID_BOARD	0x800A0002L	No board with the given nameindex available
CIFX_INVALID_CHANNEL	0x800A0003L	No channel with the given index available
CIFX_INVALID_HANDLE	0x800A0004L	Invalid handle passed to driver
CIFX_INVALID_PARAMETER	0x800A0005L	Invalid parameter
CIFX_INVALID_COMMAND	0x800A0006L	Invalid command
CIFX_INVALID_BUFFERSIZE	0x800A0007L	Invalid buffer size
CIFX_INVALID_ACCESS_SIZE	0x800A0008L	Invalid access size
CIFX_FUNCTION_FAILED	0x800A0009L	Function failed
CIFX_FILE_OPEN_FAILED	0x800A000AL	File could not be opened
CIFX_FILE_SIZE_ZERO	0x800A000BL	File size is zero
CIFX_FILE_LOAD_INSUFF_MEM	0x800A000CL	Insufficient memory to load file
CIFX_FILE_CHECKSUM_ERROR	0x800A000DL	File checksum compare failed
CIFX_FILE_READ_ERROR	0x800A000EL	Error reading from file
CIFX_FILE_TYPE_INVALID	0x800A000FL	Invalid file type
CIFX_FILE_NAME_INVALID	0x800A0010L	Invalid file name
CIFX_FUNCTION_NOT_AVAILABLE	0x800A0011L	Driver function not available
CIFX_BUFFER_TOO_SHORT	0x800A0012L	Given buffer is too short
CIFX_MEMORY_MAPPING_FAILED	0x800A0013L	Failed to map the memory
CIFX_NO_MORE_ENTRIES	0x800A0014L	No more entries available
CIFX_CALLBACK_MODE_UNKNOWN	0x800A0015L	Unkown callback handling mode
CIFX_CALLBACK_CREATE_EVENT_FAILED	0x800A0016L	Failed to create callback events
CIFX_CALLBACK_CREATE_RECV_BUFFER	0x800A0017L	Failed to create callback receive buffer

Table 49: Generic Error Codes

© Hilscher, 2009-2013

10.5.2 Generic Driver Error Codes

Error Code (Definition)	Value	Description
CIFX_DRV_NOT_INITIALIZED	0x800B0001L	Driver not initialized
CIFX_DRV_INIT_STATE_ERROR	0x800B0002L	Driver init state error
CIFX_DRV_READ_STATE_ERROR	0x800B0003L	Driver read state error
CIFX_DRV_CMD_ACTIVE	0x800B0004L	Command is active on device
CIFX_DRV_DOWNLOAD_FAILED	0x800B0005L	General error during download
CIFX_DRV_WRONG_DRIVER_VERSION	0x800B0006L	Wrong driver version
CIFX_DRV_DRIVER_NOT_LOADED	0x800B0030L	CIFx driver is not running
CIFX_DRV_INIT_ERROR	0x800B0031L	Failed to initialize the device
CIFX_DRV_CHANNEL_NOT_INITIALIZED	0x800B0032L	Channel not initialized (xOpenChannel not called)
CIFX_DRV_IO_CONTROL_FAILED	0x800B0033L	IOControl call failed
CIFX_DRV_NOT_OPENED(	0x800B0034L	Driver was not opened
CIFX_DRV_DOWNLOAD_STORAGE_UNKNOWN	0x800B0040L	Unknown download storage type (RAMFLASH based) found
CIFX_DRV_DOWNLOAD_FW_WRONG_CHANNEL	0x800B0041L	Channel number for a firmware download not supported
CIFX_DRV_DOWNLOAD_MODULE_NO_BASEOS	0x800B0042L	Modules are not allowed without a Base OS firmware

Table 50: Generic Driver Error Codes

© Hilscher, 2009-2013

10.5.3 Generic Device Error Codes

Error Code (Definition)	Value	Description
CIFX_DEV_DPM_ACCESS_ERROR	0x800C0010L	Dual port memory not accessable (board not found)
CIFX_DEV_NOT_READY	0x800C0011L	Device not ready (ready flag failed)
CIFX_DEV_NOT_RUNNING	0x800C0012L	Device not running (running flag failed)
CIFX_DEV_WATCHDOG_FAILED	0x800C0013L	Watchdog test failed
CIFX_DEV_SYSERR	0x800C0015L	Error in handshake flags
CIFX_DEV_MAILBOX_FULL	0x800C0016L	Send mailbox is full
CIFX_DEV_PUT_TIMEOUT	0x800C0017L	Send packet timeout
CIFX_DEV_GET_TIMEOUT	0x800C0018L	Receive packet timeout
CIFX_DEV_GET_NO_PACKET	0x800C0019L	No packet available
CIFX_DEV_MAILBOX_TOO_SHORT	0x800C001AL	Mailbox too short
CIFX_DEV_RESET_TIMEOUT	0x800C0020L	Reset command timeout
CIFX_DEV_NO_COM_FLAG	0x800C0021L	COM-flag not set
CIFX_DEV_EXCHANGE_FAILED	0x800C0022L	IO data exchange failed
CIFX_DEV_EXCHANGE_TIMEOUT	0x800C0023L	IO data exchange timeout
CIFX_DEV_COM_MODE_UNKNOWN	0x800C0024L	Unknown IO exchange mode
CIFX_DEV_FUNCTION_FAILED	0x800C0025L	Device function failed
CIFX_DEV_DPMSIZE_MISMATCH	0x800C0026L	DPM size differs from configuration
CIFX_DEV_STATE_MODE_UNKNOWN	0x800C0027L	Unknown state mode
CIFX_DEV_HW_PORT_IS_USED	0x800C0028L	Device is still accessed
CIFX_DEV_CONFIG_LOCK_TIMEOUT	0x800C0029L	Configuration locking timeout
CIFX_DEV_CONFIG_UNLOCK_TIMEOUT	0x800C002AL	Configuration unlocking timeout
CIFX_DEV_HOST_STATE_SET_TIMEOUT	0x800C002BL	Set HOST state timeout
CIFX_DEV_HOST_STATE_CLEAR_TIMEOUT	0x800C002CL	Clear HOST state timeout
CIFX_DEV_INITIALIZATION_TIMEOUT	0x800C002DL	Timeout during channel initialization
CIFX_DEV_BUS_STATE_ON_TIMEOUT	0x800C002EL	Set Bus ON Timeout
CIFX_DEV_BUS_STATE_OFF_TIMEOUT	0x800C002FL	Set Bus OFF Timeout
CIFX_DEV_MODULE_ALREADY_RUNNING	0x800C0040L	Module already running
CIFX_DEV_MODULE_ALREADY_EXISTS	0x800C0041L	Module already exists

Table 51: Generic Device Error Codes

© Hilscher, 2009-2013

10.6 Error Codes netX Driver

10.6.1 CIFX API Transport Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_SEND_TIMEOUT	0x800D0001L	Time out while sending data
CIFX_TRANSPORT_RECV_TIMEOUT	0x800D0002L	Time out waiting for incoming data
CIFX_TRANSPORT_CONNECT	0x800D0003L	Unable to communicate to the deviceno answer
CIFX_TRANSPORT_ABORTED	0x800D0004L	Transfer has been aborted due to keep alive timeout or interface detachment
CIFX_CONNECTOR_FUNCTIONS_READ_ERROR	0x800D0010L	Error reading the connector functions from the DLL
CIFX_CONNECTOR_IDENTIFIER_TOO_LONG	0x800D0011L	Connector delivers an identifier longer than 6 characters
CIFX_CONNECTOR_IDENTIFIER_EMPTY	0x800D0012L	Connector delivers an empty dentifier
CIFX_CONNECTOR_DUPLICATE_IDENTIFIER	0x800D0013L	Connector identifier already used

Table 52: CIFX API Transport Error Codes

10.6.2 CIFX API Transport Header State Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_ERROR_UNKNOWN	0x800E0001L	Unknown error code in transport header
CIFX_TRANSPORT_CHECKSUM_ERROR	0x800E0002L	CRC16 checksum failed
CIFX_TRANSPORT_LENGTH_INCOMPLETE	0x800E0003L	Transaction with inclomplete length detected
CIFX_TRANSPORT_DATA_TYPE_UNKOWN	0x800E0004L	Device does not support requested data type
CIFX_TRANSPORT_DEVICE_UNKNOWN	0x800E0005L	Device not availableunknown
CIFX_TRANSPORT_CHANNEL_UNKNOWN	0x800E0006L	Channel not availableunknown
CIFX_TRANSPORT_SEQUENCE	0x800E0007L	Sequence error detected
CIFX_TRANSPORT_BUFFEROVERFLOW	0x800E0008L	Buffer overflow detected
CIFX_TRANSPORT_RESOURCE	0x800E0009L	Device signals out of resources
CIFX_TRANSPORT_KEEPALIVE	0x800E000AL	Device connection monitoring error (Keep alive)
CIFX_TRANSPORT_DATA_TOO_SHORT(	0x800E000BL	Received transaction data too short

Table 53: CIFX API Transport Header State Error Codes

© Hilscher, 2009-2013

10.7 ODM Error Codes DBM V4

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_MD5_INVALID	0XC004C810	Checksum invalid
CDBM_E_INTERNALERROR	0XC004C811	Internal Error
CDBM_W_WRITEREGISTRY	0X8004C812	Error writing to the registry
CDBM_E_UNEXPECTED_VALUE_ IN_OLD_HEADER_FORMAT	0XC004C813	Error in a file containing the old DBM Header format.
CDBM_E_CHECKSUM_INVALID	0XC004C814	The Checksum of the old Header is invalid
CDBM_E_DB_ALREADY_LOADED_ FORMAT	0XC004C815	A database is already loaded
CDBM_E_NO_VALID_TRANSACTION	0XC004C816	No valid transaction handle given
CDBM_E_STD_STRUCT_ERROR	0XC004C817	An error occurred during validation of data
CDBM_E_UNSUPPORTED_ DATA_TYPE_FORMAT	0XC004C818	Unsupported DataType
CDBM_W_CLASS_DELETED_ FORMAT	0X8004C819 (Warning)	Using an Object which is marked as deleted
CDBM_W_CLIENT_DISCONNECTED	0X8004C81A (Warning)	A Client has already an outstanding connection to a Table. The connection is now destroyed.
CDBM_E_STRUCTURE_DEFINITION_ INVALID	0XC004C81B	A structure definition of an Element in a Table is invalid
CDBM_E_NO_DATA_AVAILABLE	0XC004C81C	No data available for this operation
CDBM_E_NO_VALID_STRUCTURE	0XC004C81D	No valid structure available for this operation
CDBM_E_NO_TOGGLE_STRING_ FOUND	0XC004C81E	No Toggle string found for this number
CDBM_E_ELEMENT_OUT_OF_RANGE	0XC004C81F	An element wasn’t found in the Record of a Table
CDBM_E_ELEMENT_NOT_IN_ TABLE	0XC004C820	The element is not part of the Table
CDBM_E_CANNOT_CONVERT_ INTO_CLIENT_TYPE	0XC004C821	The data can't be converted into the Client type
CDBM_E_TRANSACTION_ ALREADY_OPEN	0XC004C822	A transaction is already open. Please close this one first before opening a new one.
CDBM_I_OLD_WITHOUT_HEADER	0X4004C823 (Informational)	Use of an old DBM file Format without Header
CDBM_E_HR_FROM	0XC004C824	An HRESULT was received from a Subroutine
CDBM_E_PARAMETER	0XC004C825	A Parameter is invalid
CDBM_E_NOTIMPL	0XC004C826	Method is currently not implemented
CDBM_E_OUTOFMEMORY	0XC004C827	Out of memory
CDBM_E_NO_OPEN_TRANSACTION	0XC004C828	No transaction open
CDBM_E_NO_CONTENTS	0XC004C829	No contents available
CDBM_REC_NO_NOT_FOUND	0XC004C82A	Record not found
CDBM_STRUCTURE_ELEMENT_ NOT_FOUND	0XC004C82B	Element of the Structure not found
CDBM_E_NO_MORE_RECORDS_ IN_TABTYPE	0XC004C82C	Table type 3 can contain only one record
CDBM_E_WRITE	0XC004C82D	The data in the VARIANT must be given in a SafeArray
CDBM_E_WRITE_NO_PARRAY	0XC004C82E	The VARIANT contains no valid [parray] element

© Hilscher, 2009-2013

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_WRITE_CANT_ ACCESS_DATA	0XC004C82F	Unable to access SafeArray Data in the VARIANT
CDBM_E_WRITE_DATA	0XC004C830	To write the data of this Element it must be given as a BSTR, or as an Array of VT_UI1/VT_I1
CDBM_E_WRITE_BSTR_E1	0XC004C831	The BSTR string must have an even length.
CDBM_E_WRITE_BSTR_E2	0XC004C832	The BSTR string must contain only hex digits (0..9 and a/A..f/F).
CDBM_E_WRITE_CANT_ INTERPRET_ARRAY	0XC004C833	Unable to interpret data in the SafeArray.
CDBM_E_WRITE_VT_ERROR	0XC004C834	Data type in the SafeArray is not VT_UI1 or VT_I1.
CDBM_E_WRITE_LENGTH	0XC004C835	Data length is invalid for write operation of this type.
CDBM_WRITE_ELEMENT	0XC004C836	Element not found in the Record of the Table
CDBM_MIN_MAX_ERROR	0XC004C837	Can't write data because of min underflow or max overflow
CDBM_TABLE_EXIST	0XC004C838	Table already exist in the database
CDBM_MIN_MAX_INVALID	0XC004C839	The Min value is greater than the Max Value
CDBM_DEF_MIN_MAX_INVALID	0XC004C83A	The Default Value is not in the range between the Min value and the Max Value
CDBM_CANT_CHANGE_STRUCTURE_ WHILE_RECORDS_EXIST	0XC004C83B	It's not allowed to change the structure while Records exist in the Table
CDBM_NEW_STRUCT_NEEDS_TYPE	0XC004C83C	In a newly added structure the data type must be set also
CDBM_VALUE_ERROR	0XC004C83D	Range error while validating a value
CDBM_DATATYPE_ UNSUPPORTED_IN_RCS	0XC004C83E	The data type is unsupported in the RCS file format
CDBM_I_COUNT_OF_TABLES_ EXCEEDS_RCS_RANGE	0X4004C83F (Informational)	The count of Tables exceeds the RCS range of Tables. This can cause problems if the file is downloaded to RCS Systems
CDBM_I_COUNT_OF_TABLES_ EXCEEDS_OLDDBM_RANGE	0X4004C840 (Informational)	The count of Tables exceeds the DBM32.DLL range of Tables. This can cause problems if the file is used with older Tools using the DBM32.DLL
CDBM_UNSUPPORTED_DATATYPE_ IN_RCS_MODE	0XC004C841	The Data type is not compatible with the old database format
CDBM_WRITE_UNSTRUCTURED_1	0XC004C842	The data of an unstructured record can only be written with the 'Write' Method not with 'WriteElement'.
CDBM_READ_UNSTRUCTURED_1	0XC004C843	The data of an unstructured record can only be read with the 'Read' Method not with 'ReadElement'
CDBM_WRITE_DATA_LENGTH_ INVALID	0XC004C844	The given data length doesn’t correspond with the expected data length.
CDBM_UNKNOWN_VIEW_MODE	0XC004C845	The View Mode is unknown.
CDBM_E_DIAG_TABLE	0XC004C846	It doesn’t make much sense to add or delete records from a diagnostic table because those changes are never saved.
CDBM_E_ADR_STRING_ERROR	0XC004C847	The given Address string doesn’t fit the required format of this type where all address bytes must be in the range between 0 and FF

© Hilscher, 2009-2013

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_ERROR_FROM_VAR_ CHANGE_TYPE	0XC004C848	Function VariantChangeType return an error when trying to convert the Parameter
CDBM_E_MINERROR	0XC004C849	Error while comparing the Value with the lower range
CDBM_E_MAXERROR	0XC004C84A	Error while comparing the Value with the upper range
CDBM_E_RANGE_ERROR	0XC004C84B	Value out of Range
CDBM_E_TABLE_TYPE1	0XC004C84C	Table type 1 doesn't have a unique record length over all records
CDBM_E_TABLE_TYPE3_ ADDREC	0XC004C84D	Table type 3 doesn't allow to insert more than one Record
CDBM_E_TABTYPE1	0XC004C84E	It's not allowed to insert more Records than structure definitions in Table Type 1
CDBM_E_TOGGLE_NOT_FOUND	0XC004C84F	Could not find the string for this value in the list of valid toggle strings
CDBM_E_TOGGLE_VALUE_IS_ EMPTY_STRING	0XC004C850	The toggle string for this value is empty.
CDBM_VARIANT2BYTEARRAY_ ERROR	0XC004C851	Error during conversion of Variant to byte array
CDBM_E_SET_ELEM_PROP_ DEPENDENCY	0XC004C852	The Toggle Type needs also the additional string and the additional number entries in the Method
CDBM_E_TABTYPE1_REC_ DOESNT_CORRESPOND_ WITH_ELEMENT	0XC004C853	When reading the records of Table type 1 elementwise the record number must correspond with the element number
CDBM_TABTYPE1_NO_DATA_ FOUND_FOR_RECORD	0XC004C854	When reading the records of Table type 1 and structure definitions are present it's assumed that for each structure element a corresponding record must exist
CDBM_E_TABTYPE1_WRITE_ ELEMENT_NE_RECORD	0XC004C855	When writing the records of Table type 1 elementwise and structure definitions are present it's only allowed to write the corresponding element number in each record
CDBM_E_TABTYPE1_WRITE_ ELEMENT_NOT_FOUND	0XC004C856	When writing the records of Table type 1 with an array and structure definitions are present it's assumed that a corresponding element number of this record exist
CDBM_I_TABLE_NAME_EXCEEDS_ RCS_RANGE	0X4004C857 (Informational)	The Table name exceeds the maximum length of RCS compatible Table names
CDBM_W_CUT_STRING	0X8004C858 (Warning)	The string exceeds the maximum length and will be limited to the maximum length
CDBM_I_STRING_TOO_SHORT	0X4004C859 (Informational)	The string is below the minimum length. The minimum length will be reduced.
CDBM_I_STRING_TOO_LONG	0X4004C85A (Informational)	The string is exceeding the maximum. The maximum length will be extended.
CDBM_E_STRING_TOO_SHORT	0XC004C85B (Error)	The string is below the minimum length.
CDBM_E_STRING_TOO_LONG	0XC004C85C (Error)	The string is exceeding the maximum length
CDBM_E_WRONG_TYPE_ FOR_WRITE	0XC004C85D	Writing on the Element type with the given Data type is not implemented
CDBM_E_NO_APPEND_IN_ STRUCTURED_RECORDS	0XC004C85E	Method IDbmRecord::AppendData is not allowed for structured records

© Hilscher, 2009-2013

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_DATA_UNAVAILABLE	0XC004C85F	No data available
CDBM_E_CANT_CONVERT_ INTO	0XC004C860	Unable to convert the value into the Element type
CDBM_E_DBM_FILE_OVERFLOW	0XC004C861	You try to write a RCS like database which needs too much bytes
CDBM_E_PW_ERROR	0XC004C862	Password not correct
CDBM_E_FILELENGTH_CORRUPT	0XC004C863	The file length doesn't correspond to the length given in the Header.
CDBM_E_STRUCT_TYPE	0XC004C864	Error in the file.
CDBM_E_MD5SUM_INVALID	0XC004C865	MD5 sum invalid
CDBM_E_STRUCT_LENGTH	0XC004C866	Error in the expected and given structure length at a specific offset in the file.
CDBM_E_APPEND	0XC004C867	Append of data is only allowed if the Record contains only one data field and the field type will support this
CDBM_APPEND_NOT_ SUPPORTED	0XC004C868	Append of Data not supported by this filed type
CDBM_DATA_TYPE_APPEND_ ERROR	0XC004C869	Can't append Data of this type.
CDBM_E_UNSTRUCTURED_TABLE_ DOESNT_SUPPORT_LENGTH	0XC004C86A	A Table without structure information doesn't support a record length
CDBM_E_DISABLED_WHILE_ TRANSACTION_IS_OPEN	0XC004C86B	The Method is disabled while a transaction is open. Please close this one first and call the Method again.
CDBM_E_UNABLE_TO_CALL_ READ_ON_LINKED_LIST	0XC004C86C	The Method is disabled on a LinkedList type. Please use the IRecordCollection on this type.
CDBM_E_ELEMENT_HAS_NO_ SUBSTRUCTURE	0XC004C86D	An Element from a Table has no substructure
CDBM_STRUCT_ERROR_FROM_ VAR_CHANGE_TYPE	0XC004C86E	Error from calling VariantChangeType
CDBM_E_FOREIGNKEY_DEF	0XC004C86F	The definition of a FOREIGNKEY must contain the name of the related Table in the description and this Table must exist at this time
CDBM_E_FOREIGNKEY_ REF_TAB	0XC004C870	The description of a FOREIGNKEY must refer to a Table of type 'eDbmTableTypeLinkedList'
CDBM_E_KEY	0XC004C871	To create a Record Collection with a KEY it's necessary to have the data type KEY at the first position in all Records of the searched Table
CDBM_E_KEY_TABLE_TYPE	0XC004C872	This Method needs a Table of type 'eDbmTableTypeLinkedList'
CDBM_DATATYPE_NOT_ IMPLEMENTED	0XC004C873	This data type is currently not implemented
CDBM_INSERT_POS_NOT_ FOUND	0XC004C874	The position of the Record where the new one should be inserted wasn't found
CDBM_E_INSERT_REC_QI	0XC004C875	Error during insertion of a Record
CDBM_E_TAB_PROP	0XC004C876	Invalid Property in Table
CDBM_E_KEY_NOT_FOUND	0XC004C877	The KEY wasn't found in the Table
CDBM_E_KEY_INVALID	0XC004C878	The KEY is invalid for this operation

Table 54: ODM Error Codes DBM V4

© Hilscher, 2009-2013

11 Appendix

11.1 User Rights

User-rights are set within the FDT-container. Depending on the level the configuration is accessible by the user or read-only.

To access the **Settings**, **Configuration**, **Device Description** and **Diagnosis** panes of the DeviceNet Slave DTM you do not need special user rights. Also all users can select the decimal or hexadecimal Display mode or sort table entries.

Note: To edit, set or configure the parameters of the **Settings** and **Configuration** panes, you need user rights for *Maintenance,* for *Planning Engineer* or for *Administrator*.

The **Device Description** panes do not contain any editable elements. The indicated values in are only for information purposes.

The following tables give an overview of the user right groups and which user rights you need to configure the single parameters.

11.1.1 Settings

	Observer	Operator	Mainten-ance	Planning Engineer	Adminis-trator
Driver	D	D	X	X	X
Verify or adapt Driver Settings	-	-	X	X	X
Configuring netX Driver	-	-	X	X	X
Device Assignment	D	D	X	X	X
Scanning for Devices	-	-	X	X	X
Selecting the Device (with or without firmware)	-	-	X	X	X
Selecting the Device once more (with Firmware)	-	-	X	X	X
Firmware Download	D	D	X	X	X

Table 55: Settings (D = Displaying, X = Editing, Configuring)

11.1.2 Configuration

	Observer	Operator	Mainten-ance	Planning Engineer	Adminis-trator
General Settings	D (X)	D (X)	X	X	X
Connection Configuration	D (X)	D (X)	X	X	X
Poll Connection	D (X)	D (X)	X	X	X
Change of State Connection	D (X)	D (X)	X	X	X
Cyclic Connection	D (X)	D (X)	X	X	X
Bit-Strobe Connection	D (X)	D (X)	X	X	X
Parameter	D (X)	D (X)	X	X	X
Device Settings	D (X)	D (X)	X	X	X

Table 56: Configuration (D = Displaying, X = Editing, Configuring, D (X) = Displaying all, limited Editing or Configuring)

© Hilscher, 2009-2013

11.2 References

[1] Device Type Manager (DTM) Style Guide, Version 1.0 ; FDT-JIG - Order No. <0001-0008-000>

[2] DeviceNet Master Protocol API Manual, Revision 9, Hilscher GmbH 2013

[3] DeviceNet Slave Protocol API Manual, Revision 12, Hilscher GmbH 2013

11.3 List of Figures

Figure 1: Dialog Structure of the DeviceNet Slave DTM 13
Figure 2: Navigation Area 14
Figure 3: Status Bar – Status Fields 1 to 6 17
Figure 4: Status Bar Display Examples 17
Figure 5: Default Driver ‚cifX Device Driver' for PC cards cifX 27
Figure 6: Default Driver ‚cifX Device Driver' for PC cards cifX (example) 27
Figure 7: Default Driver ‚netX Driver' for Hilscher devices except for PC cards cifX (example) 27
Figure 8: Manual Selection of multiple drivers (Example) 28
Figure 9: netX Driver > USB/RS232 Connection 32
Figure 10: netX Driver > TCP Connection 35
Figure 11: Device Assignment - detected Devices (* The name of the device class is displayed.) – Example for a device without firmware 37
Figure 12: Device Assignment - detected Devices (* The name of the device class is displayed.) Example for Devices without Firmware 39
Figure 13: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for a device without firmware / one Device is selected 40
Figure 14: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for Devices with and without Firmware / one Device is selected 41
Figure 15: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for a device with firmware / one Device is selected 42
Figure 16: Firmware Download 43
Figure 17: Window ‘Select Firmware File' (Example CIFX) 44
Figure 18: Request Select Firmware File - Example 46
Figure 19: Firmware Download – Download 46
Figure 20: Firmware-Download –Question 46
Figure 21: Clock Symbol and Hook Symbol green 47
Figure 22: Firmware Download - Progress Bar 47
Figure 23: General Settings - Attributes of the device identification (Slave DTM at the Root-Line (Stand-Alone Slave)) 51
Figure 24: General Settings - Attributes of the device identification (Slave DTM at the Master busline) 51
Figure 25: General Settings - Verify Device ID 52
Figure 26: General Settings - Enable Address Switch (Slave DTM at the Root-Line (Stand-Alone Slave, Exmaple COMX 10XX-DNS/DNS)) 52
Figure 27: Configuration Dialog Connection 53
Figure 28: Indication of possible combinations of connection types 53
Figure 29: Warning invalid connection type combination 53
Figure 30: Poll Connection - Consumption and Production 54
Figure 31: Poll Connection – Timing 54
Figure 32: Change of State Connection - Consumption and Production 56
Figure 33: Change of State Connection – Timing 56
Figure 34: Cyclic Connection - Consumption and Production 58
Figure 35: Cyclic Connection - Timing 58
Figure 36: Bit-Strobe Connection - Consumption and Production 59
Figure 37: Bit-Strobe Connection - Timing 59
Figure 38: Parameter Configuration 61

© Hilscher, 2009-2013

Figure 39: Configuration > Device Settings 62
Figure 40: Device Settings > Start of Bus Communication 63
Figure 41: Device Settings > Application Monitoring 63
Figure 42: General Diagnosis 69
Figure 43: Firmware Diagnosis (*The name of the Firmware is displayed.) 71
Figure 44: Extended Diagnosis > [Folder Name] > Task Information Example Display 73
Figure 45: Extended Diagnosis > [Folder Name] > IniBatch Status Example Display 74
Figure 46: Extended Diagnosis > DNS_FAL > DNS Common Status 75
Figure 47: Extended Diagnosis > DNS_FAL > AP Commands Counter 76
Figure 48: Extended Diagnosis > DNS_FAL > CAN Command Counter 76
Figure 49: Extended Diagnosis > DNS_FAL > Timer Counter 77
Figure 50: Extended Diagnosis > CAN_DL > AP Commands Counter 78
Figure 51: Extended Diagnosis > CAN_DL > CAN Driver Status 79
Figure 52: Packet Monitor 81
Figure 53: Send > Packet Header and Send Data 82
Figure 54: Packet Header and Receive Data 83
Figure 55: Packet Description GET DPM IO INFO REQUEST 84
Figure 56: Example - Reading data via „Get DPM IO Info“ – Send > Packet header 85
Figure 57: Example - Read data via „Get DPM IO Info“ - Send > Send Data 85
Figure 58: Packet Description GET DPM IO INFO CONFIRMATION 86
Figure 59: Example - Reading data via „Get DPM IO Info“ – Receive > Packet header 87
Figure 60: Example - Reading data via „Get DPM IO Info“ – Receive > Receive data 88
Figure 61: IO Monitor 89

11.4 List of Tables

Table 1: Descriptions Dialog Panes 5
Table 2: General Device Information 14
Table 3: Overview Dialog Panes 15
Table 4: OK, Cancel, Apply and Help 16
Table 5: Selecting, inserting, deleting Table Line 16
Table 6: Status Bar Icons [1] 17
Table 7: Getting started - Configuration Steps (Slave DTM at the Root-Line (Stand-Alone Slave)) 20
Table 8: Getting started - Configuration Steps (Slave DTM at the Master busline) 23
Table 9: Descriptions of the Dialog Panes Settings 24
Table 10: Driver Selection List Parameters 27
Table 11: Parameters netX Driver > USB/RS232 Connection 33
Table 12: Parameters netX Driver > TCP Connection 36
Table 13: Parameters of the Device Assignment 38
Table 14: Parameter Firmware Download 43
Table 15: Parameters Select Firmware File 45
Table 16: Dialog Panes Configuration 48
Table 17: General Settings - Attributes of the device identification 51
Table 18: General Settings - Enable Address Switch 52
Table 19: Possible connection types 53
Table 20: Change Parameter Data 61
Table 21: Range of Value / Value for the Watchdog time 63
Table 22: Descriptions of the Dialog Panes Device Description 64
Table 23: General Device Information 64
Table 24: Device Description – EDS Viewer 65
Table 25: Descriptions of the Diagnosis Panes 68
Table 26: Indication General Diagnosis 70
Table 27: Parameter General Diagnosis 70

© Hilscher, 2009-2013

Table 28: Description Table Task Information 71
Table 29: Descriptions of the Dialog Panes Extended Diagnosis 72
Table 30: Extended Diagnosis > [Folder Name] > Task Information 73
Table 31: Extended Diagnosis > [Folder Name] > IniBatch Status 74
Table 32: Extended Diagnosis > DNS_FAL > DNS Common Status 75
Table 33: Extended Diagnosis > DNS_FAL > AP Commands Counter 76
Table 34: Extended Diagnosis > DNS_FAL > CAN Command Counter 76
Table 35: Extended Diagnosis > DNS_FAL > Timer Counter 77
Table 36: Extended Diagnosis > CAN_DL > AP Commands Counter 78
Table 37: Extended Diagnosis > CAN_DL > CAN Driver Status 79
Table 38: Descriptions of the Diagnosis Panes 80
Table 39: Descriptions Packet Header 82
Table 40: Descriptions Packet Header 83
Table 41: Overview Error Codes and Ranges 91
Table 42: RCX General Task Errors 92
Table 43: RCX Common Status & Errors Codes 93
Table 44: RCX Status & Error Codes 94
Table 45: RCX Status & Error Codes Slave State 94
Table 46: ODM Error Codes - General ODM Error Codes 95
Table 47: ODM Error Codes - General ODM Driver Error Codess 96
Table 48: cifX Driver Specific ODM Error Codes 99
Table 49: Generic Error Codes 100
Table 50: Generic Driver Error Codes 101
Table 51: Generic Device Error Codes 102
Table 52: CIFX API Transport Error Codes 103
Table 53: CIFX API Transport Header State Error Codes 103
Table 54: ODM Error Codes DBM V4 107
Table 55: Settings (D = Displaying, X = Editing, Configuring) 108
Table 56: Configuration (D = Displaying, X = Editing, Configuring, D (X) = Displaying all, limited Editing or Configuring) 108

11.5 Glossary

Data Packets

Data packets, i. e. messages are self-contained blocks of defined data length. The packets are used to communicate with the firmware and they are exchanged between the application (configuration software) and the firmware in the device. Using the configuration software feature *Packet Monitor* Packets can be sent once or cyclically to the connected device controlled by the user and packets received can be displayed.

Data packets comprise from a **Packet Header** and the **Send Data** or from a **Packet Header** and the **Receive Data**. The packet header can be evaluated by the receiver of the packet and contain the sender and receiver address, the data length, an ID number, status and error messages and the command or response code. The minimum packet size amounts 40 Byte for the packet header. The sending and receiving data are added.

DTM

Device Type Manager

© Hilscher, 2009-2013

The Device Type Manager (DTM) is a software module with graphical user interface for the configuration and/or for diagnosis of devices.

EDS

An Electronic Data Sheet (EDS) provides information necessary to access and alter the configurable parameters of a device. An Electronic Data Sheet (EDS) is an external file that contains information about configurable attributes for the device, including object addresses of each parameter. The application objects in a device represent the destination addresses for configuration data. These addresses are encoded in the EDS.

FDT

Field Device Tool

FDT specifies an interface, in order to be able to use DTM (Device Type Manager) in different applications of different manufacturers.

MAC ID

MAC = Media Access Control

The network address of a device serves to distinguish itself on a DeviceNet fieldbus system from any other device or Slave on this network. This should be a unique number for each device.

Master

Master devices initiate the data exchange at the bus.

ODMV3

The Online-Data-Manager Version 3 (ODMV3) is an application interface. The ODMV3 works as a server, which can be run as an out-proc server or system service. Its task is to provide different applications (e. g. SYCON.net), access to multiple devices and even share one device amongst several applications.

Slave

Slave devices are configured by the Master and perform then the communication.

© Hilscher, 2009-2013

11.6 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
New Delhi - 110 065
Phone: +91 11 26915430
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com

© Hilscher, 2009-2013